The Dissertation Committee for Ali Raza
certifies that this is the approved version of the following dissertation:

Immunity-based Framework for Heterogeneous Mobile Robotic Systems

Committee:

Benito R. Fernández, Supervisor

Risto Miikkulainen

Matthew I. Campbell

Dragan Djurdjanovic

Luis Sentis

Immunity-based Framework for Heterogeneous Mobile Robotic Systems

by

Ali Raza, B.E., M.S.

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

December 2012

Dedicated to my children: Abdullah and Hassaan.

Acknowledgments

I would like to express my heartfelt gratitude to Dr. Benito R. Fernández for his continuous support and encouragement. Truly, he has been a mentor and a guide. Not only he introduced me to the field of artificial immunity, he helped me to mature into a plasma cell. I am also deeply grateful to Fulbright Scholarship Program for their enabling support throughout the course of study. I would also like to thank my dissertation committee of Risto Miikkulainen, Matthew I. Campbell, Dragan Djurdjanovic and Luis Sentis for their input and guidance.

My Tony Robbins is Khalid Arif, a boilermaker and a friend indeed. He pushed me hard to pursue higher studies. I am also indebted, both literally and figuratively, to Samsoon Inayat whose reinforcement has always been there for me. I am also grateful to Tehmoor Dar whose support and encouragement helped me through the good times and bad.

I would also thank my parents who prayed their hearts out, my siblings who cheered me up and my wife who stood by me. Their unconditional love makes me realize that a doctoral degree pursuit is not an individual endeavor but a group effort.

Immunity-based Framework for Heterogeneous Mobile Robotic Systems

Publication No. _____

Ali Raza, Ph.D.
The University of Texas at Austin, 2012

Supervisor: Benito R. Fernández

Artificial immune systems (AIS), biologically inspired from natural immune functions, can be reactive as well as adaptive in handling generic and varying pathogens, respectively. Researchers have used the immunological metaphors to solve science and engineering problems where unknown/unexpected scenarios are plausible. AIS can be a suitable choice for various robotic applications requiring reactive and/or deliberative control. This research aims to translate modern trends in immunology, to develop an immunity-based framework, to control a team of heterogenous robots on varying levels of task allocation and mutual interactions. The presented framework is designed to work as a multi-agent system in which safe environment is treated reactively through innate immunity, whereas unsafe situations invoke adaptive part of immune system, simultaneously. Heterogeneity is defined in terms of different sensing and/or actuation capabilities as well as in terms of different behavior-sets

robot(s) possess. Task allocation ranges from primitive to advanced behaviors. Mutual interactions, on the other hand, range from simpler one-to-one interaction to mutual coordination. In this context, a new immunity-based algorithm has been developed & tested, combining innate and adaptive immunities, to regulate cell populations and corresponding maturations, along with internal health indicators, in order to effectively arbitrate behaviors/robots in a heterogenous robotic system, in environments that are dynamic and unstructured.

Table of Contents

Acknowledgments v

Abstract vi

List of Tables xii

List of Figures xiii

Chapter 1. Introduction 1

1.1 Motivation and Background 1
1.2 Objectives and Approach 4
1.2.1 Research Objectives 4
1.2.2 Approach 9
1.3 Dissertation Outline 11

Chapter 2. Background and Related Work 13

2.1 Biological Immunity 13
2.1.1 Innate Immunity 15
2.1.2 Adaptive Immunity 16
2.2 Computational Interpretation of Immunity 18
2.3 Review of Immuno-inspired Robotic Applications 22
2.3.1 Robotic Applications using Clonal Selection 22
2.3.2 Robotic Applications using Idiotypic Network 25
2.3.2.1 Ishiguro-Watanabe Stream 26
2.3.2.2 Whitbrook's Stream 30
2.3.2.3 Lee-Sim Stream 31
2.3.2.4 Li-Wang Stream 31
2.3.2.5 Luh's Work 32
2.3.2.6 Non-Farmer Approach 34

2.3.3 Robotic Applications using Danger Theory 35
2.4 Literature Findings and Conclusions 36
2.4.1 On using AIS 37
2.4.2 On immuno-inspired robotics 43
2.5 Relationship between Dissertation Research and Prior Work . 46
2.5.1 Computational cost of antibody concentrations 46
2.5.2 Robot-Heterogeneity 49
2.6 Chapter Summary 50

Chapter 3. Innate Component of the Framework 51
3.1 Inspiration from Innate Immunity 51
3.2 Mathematical Abstraction 53
3.2.1 Representation 54
3.2.2 Phagocytosis 57
3.2.2.1 Chemotaxis 58
3.2.2.2 Biased Random Walk 59
3.2.3 Inflammation 61
3.2.4 Dendritic cells 63
3.2.5 Initiation of T-cell Maturity 66
3.3 Implementation of Innate Immunity Module 66
3.3.1 Algorithm 69
3.3.2 Iteration count of Monte Carlo simulation 69
3.4 Chapter Summary 72

Chapter 4. Adaptive Component of the Framework 76
4.1 Inspiration from Adaptive Immunity 76
4.2 Mathematical Abstraction 79
4.2.1 Representation 79
4.2.2 T Lymphocytes 83
4.2.3 B Lymphocytes 85
4.2.3.1 Clonal Selection 85
4.2.3.2 Idiotypic Network 86
4.2.4 Immune-memory 88

4.3 Implementation of Adaptive Immunity Module 89
4.3.1 Algorithm 89
4.4 Chapter Summary 89

Chapter 5. Experimentation **93**
5.1 Scenarios 94
5.1.1 Shaped-world Scenarios 95
5.1.2 Mapped-world Scenarios 96
5.1.3 Maze-world Scenarios 98
5.1.4 Distributed-world Scenarios 99
5.2 Robot configurations 102
5.3 Evaluation 104
5.4 Simulator 106
5.5 Chapter Summary 108

Chapter 6. Results and Discussion **110**
6.1 Single-Robotic Systems 110
6.1.1 SAR in a mapped-world 110
6.1.2 Navigation through a maze 113
6.1.3 Single robot in a distributed-world 118
6.2 Heterogeneous Mobile Robotic Systems 121
6.2.1 HMRS in a shaped-world 121
6.2.2 HMRS in a distributed-world 127
6.3 Computational Environment 129
6.4 Comparison with Other Robotic Approaches 130
6.5 Chapter Summary 133

Chapter 7. Conclusions and Future Directions **135**
7.1 Conclusions 135
7.2 Future Directions 138

Appendices **140**

Appendix A. Details of Immuno-inspired Robotic Applications **141**

Appendix B. List of Symbols **149**

Appendix C. List of Abbreviations **150**

Index **151**

Bibliography **153**

List of Tables

3.1 Metaphors for Innate Component of Immunity-based HMRS . 52
3.2 Algorithm for innate component of immunity for HMRS . . . 71

4.1 Metaphors for Adaptive Component of Immunity-based HMRS. 77
4.2 Algorithm for adaptive component of immunity for HMRS . . 90

5.1 Utility of shaped-world scenarios 96
5.2 Utility of mapped-world scenarios 97
5.3 Utility of maze-world scenarios 99
5.4 Utility of distributed-world scenarios 101
5.5 Robot configurations used in SAR experimentation 103

6.1 SAR results in mapped world scenarios 112
6.2 Results of robot navigation through maze-world scenarios . . . 118
6.3 SAR results in shaped world scenarios 125
6.4 SAR results in distributed world scenarios 129
6.5 Comparison between IBF and Reactive Immune Network of Luh and Liu [50] 133

A.1 Robotic Applications using Clonal Selection Theory 141
A.2 Mathematical Details of CS-based Robotic Applications 142
A.3 Robotic Applications using Idiotypic Network Theory 143
A.4 Representation Scheme of IN-based Robotic Applications . . . 145
A.5 Mathematical Details of IN-based Robotic Applications 147

List of Figures

1.1 Schematic diagram of a search and rescue system using immunity-based framework for heterogeneous mobile robotic system. The three robots shown may have different sensors, actuators, brains (processors), power sources, even communication media and protocols. They may have different perceptions and conceptions of the world. 12

2.1 An overview of biological immune system (BIS). In BIS, different cells and mechanisms generate a defensive stance against foreign pathogens. Key components are shown for innate and adaptive immunity mechanisms. 14

2.2 Genealogical Chart of AIS-based Robotic Applications. 23

2.3 AIS-based framework for problem solving, adapted from Castro [15]. 38

2.4 A review of immuno-inspired robotic applications 47

2.5 Time of computation with respect to increasing number of antibodies (A_i) in a idiotypic-network-only simulation 48

2.6 Genealogy of robot's heterogeneity 49

3.1 A generalized structure of innate immunity in a robot 53

3.2 Structure of bacterial representation. The illustrated robot has a $\left\{[-\pi/3, \pi/3],\ 16,\ \pi/24\right\}$ configuration. 55

3.3 The robot, with $\left\{[-\pi/3, \pi/3],\ 16,\ \pi/24\right\}$ configuration, avoids the obstacle (black) and seeks the target (gray) using monokines as steering agents. 61

3.4 Details of innate immunity for robot navigation. (a) The trace showing obstacle-avoidance and target-seeking behaviors. (b) The steering direction towards the maximum probability density of monocytes. (c) Current and past positions in the trace. (d) Closeup of bacterial invasion in the robot. 62

3.5 Multiple simulation runs of innate component of immunity with same robot-configuration and number of monokines 68

3.6 Multiple simulation runs of innate component of immunity with different robot-configurations and number of monokines 70
3.7 Symmetric and skewed scenarios with different neighborhoods 73
3.8 Results of test-function to calculate the number of required Monte Carlo iterations for a skewed scenario 74
3.9 Results of test-function to calculate the number of required Monte Carlo iterations for a symmetric scenario 75

4.1 A generalized structure of robot's AIS in an HMRS, combining innate and adaptive components 78
4.2 Structure of epitopes in antigen representation. The illustrated robot, with $\{[-\pi/3, \pi/3], 14, \pi/21\}$ configuration, is fitted with photosensors as localizing and color-sensors as speciality sensors. 81
4.3 Structure of paratopes and idiotopes in hybrid idiotypic network. In this illustration, number of bits in fixed length robot-ID ($N_{bits_{R_i}}$) is 4, number of bits in flexible length binary string (N_a) is 14 and $i, j = 1, \ldots, N_a$. 82
4.4 Structure of Antigen-Antibody and Antibody-Antibody interactions in an idiotypic network. 86
4.5 (a) The robot, with $\{[-\pi/3, \pi/3], 16, \pi/24\}$ configuration, on its way to avoid the obstacle (black) and seek the target (gray). (b) The sensor reading at last instance 91
4.6 Antibody concentrations of the robot, with $\{[-\pi/3, \pi/3], 16, \pi/24\}$ configuration, during one instance of its motion 92
4.7 (a) Symmetric starting scenario, (b) Skewed starting scenario, illustrating the possibility of skipping the target 92

5.1 Different shaped-world scenarios 95
5.2 Different mapped-world scenarios: (a) & (b) from Borenstein et al. [10], (c) from Ulrich and Borenstein [78], (d) from Minguez and Montano [57] and (e) & (f) from Fernández et al. [25] . . 98
5.3 Different maze-world scenarios 100
5.4 Different distributed-world scenarios 102
5.5 Antigen mapping with different robot morphologies 105

5.6 One instance of simulation involving two robots with different sensing capabilities. Different starting positions are also shown in a distributed-world scenario. 107
5.7 User interface of the simulator output. Shown are the virtual environment (left), graphs for Energy, Collisions, Inflammation, and Immunity. The pink bars at x and y axis indicate the maturity of immune-response and its depth, respectively. In the virtual environment, traces of where the robots (2 in this case) have been. 109

6.1 Simulations in mapped-world scenarios with different robot-morphologies 111
6.2 Performance evaluation in mapped-world scenarios 112
6.3 Navigation through a 12x12 maze with performance indicators 114
6.4 Different stages of robot's movement through a 16x16 maze. . 116
6.5 Illustration of framework's ability to navigate through increasingly difficult mazes. 117
6.6 Performance evaluation in maze-world scenarios 118
6.7 One robot searches three targets in a distributed-world scenario 119
6.8 Performance indicators corresponding to single-robotic navigation in a distributed-world scenario 120
6.9 SAR with single robot with different starting positions in a 12-3 Obstacle-target arena configuration 120
6.10 Shaped-world scenario (E) with different stages in a simulation run 123
6.11 Multiple simulations in a shaped-world scenario (E) with different starting positions 124
6.12 Different shaped-world scenarios for a two-robot SAR task . . 124
6.13 Performance evaluation in shaped-world scenarios 125
6.14 Performance indicators during one simulation run: (a). Antibody concentrations in one iteration with zero initial conditions, (b). Squashing function for antibody concentrations, (c). Energy trace, (d). Collision trace. 126
6.15 Simulation runs with increasing obstacle/target density in distributed-world scenarios 128
6.16 Simulation screenshots of various distributed-world scenarios and starting conditions 128
6.17 Performance evaluation in distributed-world scenarios 130

6.18 Performance of VFH-method in different shaped-world scenarios indicating its limitations w.r.t the performance of IBF in Fig. 6.12 132
6.19 Performance of RIN [50] in a maze world indicating its only success in a 10x10 maze 133

Chapter 1

Introduction

1.1 Motivation and Background

Mobile robots present themselves as an embodied manifestation of artificial intelligence by performing various tasks in unknown or partially known environments. Mobile robotic applications started their journey with a single robot, performing simple navigation tasks, and gradually developed into multi-robotic systems that exhibit swarm intelligence in some cases [66] and heterogeneous robotic activity in others [5, 76, 82]. Other synchronistic research areas include, but are not limited to, conflict resolution in multiple behaviors [64, 67], probabilistic robotics [73] and behavior evolution [41].

This research is aimed to explore robot-heterogeneity and consequent problem of designing a framework that can handle robots of different capabilities, over and above the classic problem of navigation in unknown environments. The need of heterogeneity arises from a premise that a robot may not perform all the tasks because it cannot be equipped with all the necessary sensory/actuarory capabilities, in a multi-robot system. A *specialist* robot [65], equipped with costly hardware, can be employed whenever a situation calls for it, whereas a less expensive robot can be expendable. A roboticist, in other

words, would prefer to send some hoplites for a minor skirmish rather than soaking her Hercules' robe with Nessus's blood[1]. Examples of *heterogeneous mobile robotic systems* (HMRS) can be numerous e.g. a bomb disposal system in which a scanning-robot is fitted with sensors to sniff bombs, whereas a diffuser-robot has the actuation to dispose them off [33]. Similar is the case of a search-and-rescue (SAR) system, the one experimented in this research, in which a rescue robot has a gripper along with a minimalistic sensory configuration, whereas a search robot has no actuation other than the navigation capability but has all the necessary sensors to search a target. Sometimes a homogeneous robotic system can incidentally become a heterogeneous one when a robot suffers a partial hardware breakdown. Robot-heterogeneity can, therefore, be a consequence of multiple factors e.g. task-heterogeneity, cost considerations, hardware variations, etc. It can also present itself in either morphological, behavioral or incidental avatars. In a quest for developing a framework for HMRS, an *artificial immune system* (AIS) is implemented because it enables a cooperation among morphologically different cells, in a structured manner, by maintaining a homeostasis. It can also, like a biological immune system, handle unknown pathogens adaptively and exhibit intelligence in terms of self-organization, learning, adaptation, and scalability [19].

The literature on immuno-inspired robotic applications indicates that researchers have employed either single robot or multiple robots of the same type in their research (see chapter 2 for an exhaustive literature review). Even

[1] Ovid, Metamorphoses, IX l.132-3

in the cases of Li and Wang [47] and Duan et al. [23], where variants of predator-prey experiment are implemented, no morphological distinction is made between the robots. In classical predator-prey experiments, on the other hand, a predator is embedded with higher sensitivities and a prey is modeled with higher actuation capabilities. For example, a predator may be given a better vision, whereas a prey may have higher speeds to evade an attack. It is, therefore, important for an HMRS to account for robot taxonomy.

It is also observed that immunity-based robotic applications have used older immune models (e.g. *idiotypic network theory* [38]). Recent research in immunology, however, is more focused on *innate immune system* and its interactions with adaptive immunity (e.g. *danger theory* [53]). Moreover, the reported research does not employ the concepts of chemotaxis, phagocytosis and inflammation, whereas antigen-presentation, T-cell functionality and immune-memory are rarely used. The ability of monocytes to respond to invading pathogen, regulate their populations and maintain an internal state of health can be very useful to respond to new situations, arbitrate behavior modules and maintain robot integrity, respectively, in a robotic system. The central idea of this research, therefore, stems from this need to use modern immunological definitions to exhibit intelligence in terms of a robotic embodiment.

In this context, an immunity-based framework (IBF) is designed which applies modern immuno-definitions to control heterogeneous robotic systems. It uses *innate immunity* when environment is contextualized as *safe* and re-

sorts to *adaptive immunity* when considered *dangerous* while regulating internal health indicators like inflammation. The IBF is an *all-encompassing* framework that is reactive as well as adaptive, non-deterministic as well as scalable.

1.2 Objectives and Approach

Initially, the innate component of IBF navigates each robot in the arena by following a monocytic walk, distributively. Inflammation levels may rise or drop on the basis of robots' experiences. The resulting maturity of dendritic cells along with the gradient of chemoattractants/chemorepellents in robot's vicinity may call adaptive immunity to respond. The adaptive component, in return, follows a T-cell functionality first and B-cell functionality later to resolve the situation. At this stage, the IBF selects different robots on the basis of clonal selection theory [13], while performing distributed tasks on the basis of self non-self [11] and idiotypic network theory [38], albeit robot-heterogeneity.

1.2.1 Research Objectives

The research objectives are as under:

1 *To test the hypothesis that an immunity-based framework, combining innate and adaptive components, is more effective than the conventional idiotypic network approach for heterogeneous mobile robotic systems, performing in unstructured environ-*

ments.

Literature indicates that roboticists have not previously considered innate immunity because it has been a mystery until recent discoveries in this field. This research aims to include innate immunity in such a manner that adaptive immunity is called only when required; unlike the conventional approach that keeps on running a clonal-selection/immune-network algorithm, all the time. Innate immunity has two major components: one to call cells to handle general pathogens and second to co-stimulate adaptive immunity.

2 *To incorporate functions of innate immunity for reactive robot tasks.*

It is observed that phagocytosis is not incorporated in any of the immunity-based robotic applications. Phagocytosis is the first cellular response to invading pathogens within the realm of innate immunity [72]. This involves engulfing of solid particles by the cell membrane. This process should be investigated as a possible reactive response using a population based stochastic implementation.

2.1 *To implement phagocytosis in order to exhibit a population based reactive response to chemoattractants of invading bacterium.*

Danger theory of immunology [53] states that the process of antigen presentation is initiated on the basis of alarm signals from stressed cells that

results in maturity of dendritic cells. This phenomenon should be abstracted to contextualize the environment. Dendritic cells can be immature, semi-mature or mature on the basis of danger/alarm signals during robot navigation. Danger signals can be abstracted from either sensory data or internal inflammation functions. The antigenic data is then contextualized as safe or unsafe and communicated to the adaptive immune system.

2.2 *To perform maturation of dendritic cells on the basis of either external, internal or both alarm signals.*

2.3 *To present contextualized antigenic data to adaptive immune system.*

It is also desired to implement a deeper biological inspiration because a single aspect of AIS may not be sufficient to incorporate a successful robotic system. A robot may encounter situations where the reactive approach of innate immunity is insufficient e.g. local minima trapping or where a different robot is required to handle the situation e.g. rescue scenario. It is important to highlight here that an AIS can be *all-encompassing*. It has functions that provide a distributed network structure like idiotypic network, reinforcement learning like T-cell algorithms, evolution like somatic hypermutation, short term learning like metadynamics and weighted sum of attractive/repulsive forces like dendritic cell algorithms. A two-layered approach can be one of the solutions where one layer corresponds to antigenic data and the other to environment-contextualization in terms of safe or dangerous signals.

3 *To implement adaptive immunity in order to exhibit an antigen-specific response*

3.1 *To mature T-cells on the basis of co-stimulation from mature dendritic cells. T-cell algorithm should be abstracted in terms of an adaptive critic as well as an effector.*

3.2 *B-cell maturity should result in plasma cells and memory-B-cells.*

3.2.1 *Plasma cells should be able to evolve and regulate antibodies according to clonal selection theory.*

3.2.2 *To constitute an immune-memory with matured B and T lymphocyte populations.*

The benefit of using robots as an application is in their embodiment because the information of sensor locations and system dynamics can be a part of representation schema. It is, therefore, necessary to design an idiotypic representation scheme that ensures robot heterogeneity.

3.3 *To design idiotypic network in such a manner that robot heterogeneity can be translated in terms of paratope-idiotope representation.*

Evolved antibodies assist in microbial destruction by binding to them and making them available to phagocytes and consequent complement cascading. This mechanism is an antigen-specific response of the adaptive immune system.

3.4 *To develop a communication setup in which antibodies with higher concentrations are communicated back to the respective site of pathogen invasion.*

This concludes our discussion on defining research objectives corresponding to AIS. It is appropriate now to define research objectives that are related to robotics. Selection of a particular robotic application is also important. Robot taxonomy is also an important aspect to be considered as well; especially when one aims to develop a generic framework for a number of (heterogeneous) robot platforms. It has also been identified that benchmark problems should also be considered and tested to validate the proposed algorithms.

4 *To test different scenarios in order to establish validity of the proposed framework. Search and Rescue application can be chosen as a representative of a heterogeneous mobile robotic system.*

4.1 *Search robots should have a different set of hardware configuration, in terms of sensory and/or actuatory components, than that of rescue robots.*

4.2 *To develop a test environment that is unstructured.*

Reported applications also limit themselves in terms of using predefined behaviors to arbitrate from. This poses a problem in heterogeneous mobile robotic systems because each robot would then require programming

of different behavioral modules ahead of time. Ideally, intelligence should emerge irrespective of hardware configuration of robots. This leads us to opt for behavior evolution rather than conventional behavior arbitration because coupling antibodies to predefined actions or behavior-modules stops inclusion of new behaviors.

5 *To implement an evolutionary mechanism on the basis of clonal selection to evolve antibody-paratopes.*

5.1 *To test that clonal selection based evolutionary mechanism is useful to find a suitable robot from a team of heterogeneous robots.*

Multiple runs in different test scenarios are required to test the effectiveness of proposed methodology against other techniques. Internal functions of the proposed AIS methodology also provide a qualitative analysis of its effectiveness like a reduced inflammation resulting from a successful rescue.

6 *Proposed framework must be tested to establish its effectiveness in comparison with other well known immunity-based application(s), against metrics of computational cost, task completion and adaptability.*

1.2.2 Approach

A *Search-and-Rescue* system is selected as a representative heterogeneous robotic system because a number of different robotic tasks and behaviors can be tested. Each robot in such a system demands some basic

tasks like reactive navigation, obstacle avoidance and target seeking, and some additional/speciality tasks like detection or rescue capabilities. Therefore, a framework that can successfully "*search-and-rescue*" also exhibits that it can navigate different robotic platforms. It can also be scaled to include more robots in either roles. Different search strategies can be tried e.g. radial, random, etc. Similarly, rescue task can require two or more robots to push a heavy load together, thus requiring a continuous communications to establish cooperation.

A schematic diagram of one possible configuration of SAR system is presented in fig. 1.1. It shows that each robot has an innate immunity module that performs chemotaxis, phagocytosis and/or antigen presentation on the basis of local sensory data. Output mapping is done on the basis of evoked immunity functions. If dendritic cells within a robot mature on the basis of accumulated danger signals, they output co-stimulation signals through the on-board communication module. This results in invocation of adaptive immunity and cause T-cell maturity. Adaptive immunity, also known as cell-mediated immunity by some, can transmit evolved/stimulated antibodies back to the first robot, in one possible situation. Other possibility can invoke cytotoxic T-cells. The antibodies from B-cells trigger an actuation response and, resultantly, resolve the dangerous situation that initiated the dendritic cell maturity. The arena in which robots have to perform different tasks, also has randomly positioned targets and obstacles.

1.3 Dissertation Outline

The current chapter discussed a brief introduction. The remainder of this dissertation is structured as under:

Chapter 2 presents a background of immunological inspirations and a comprehensive review of literature on immuno-inspired robotics. It also identifies the voids to avoid in the development of IBF.

Chapter 3 presents the details of innate component of IBF along with its mathematical abstractions. Implementation issues are also discussed along with the final algorithm of the component.

Chapter 4 shows the details of adaptive component of IBF and its mathematical abstractions. It is followed by the implementation details and final algorithm of the component.

Chapter 5 provides the details of experimentation. Different scenarios, robot configurations, starting conditions and evaluation criteria are also discussed.

Chapter 6 shows the results and discusses the experiments.

Chapter 7 draws the conclusions of this dissertation. A summary of contributions and future directions is also presented.

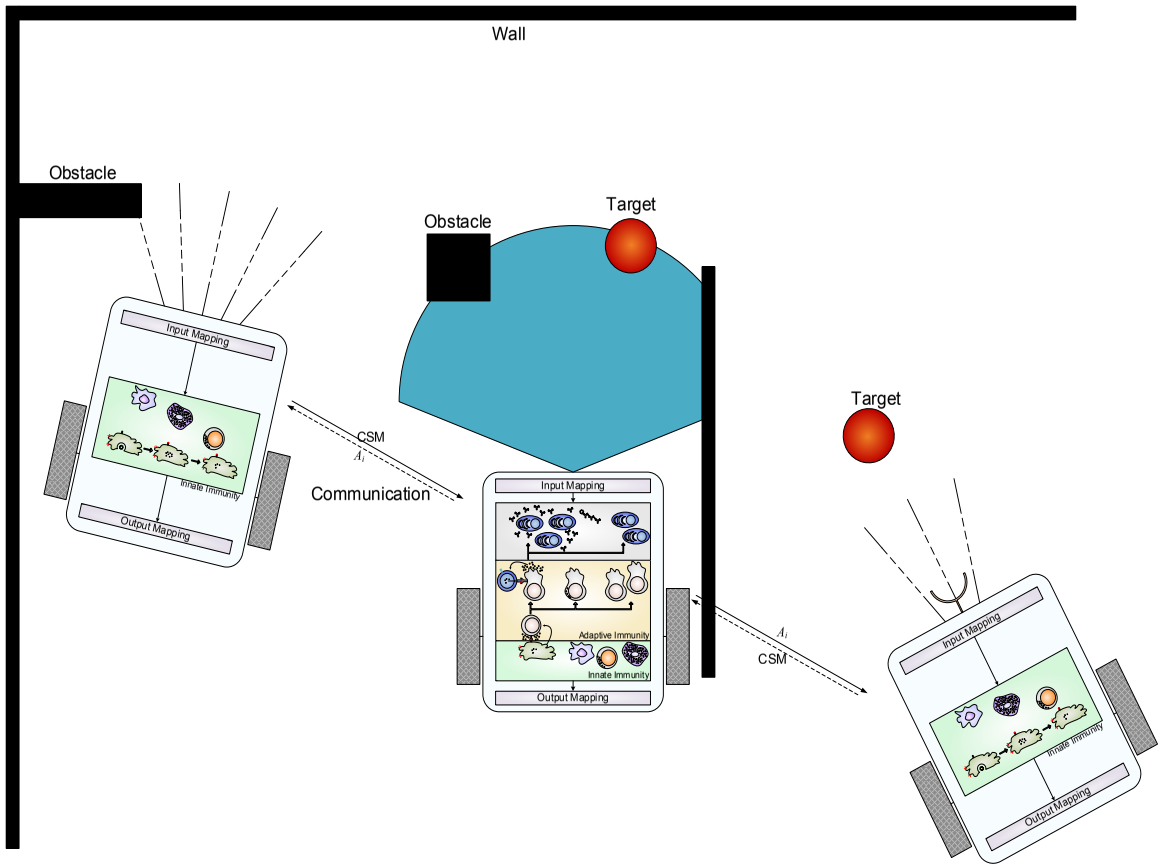


Figure 1.1: Schematic diagram of a search and rescue system using immunity-based framework for heterogeneous mobile robotic system. The three robots shown may have different sensors, actuators, brains (processors), power sources, even communication media and protocols. They may have different perceptions and conceptions of the world.

Chapter 2

Background and Related Work

An AIS is not a *replica* of a biological immune system (BIS) but only an inspiration. Therefore, only the relevant biological information is detailed in the following section. Subsequent sections transfer the basics of BIS-working into the computational realm. It is followed by a critique of immuno-inspired robotic applications which is aimed to identify the problematic areas, in view of modern trends in both robotics and immunology. The findings are later used in the development of the Immune-Based Framework (IBF).

2.1 Biological Immunity

Biological immunology is a modern research area. It is evident from 2011 Nobel Prize for Physiology or Medicine awarded to Bruce A. Beutler, Jules A. Hoffmann and Ralph M. Steinman[1] on their discoveries related to innate immunity and dendritic cells, respectively. The methodology presented in this research (chapters 3 & 4) is inspired from these modern trends in immunology, including a more recent *danger theory*. Although immunity in non-vertebrates is known, this brief is limited to that of vertebrates only because

[1] "Nobel Prizes 2011". Nobelprize.org. 13 Aug 2012

of the extent of respective theoretical explanations, duly backed by empirical evidences.

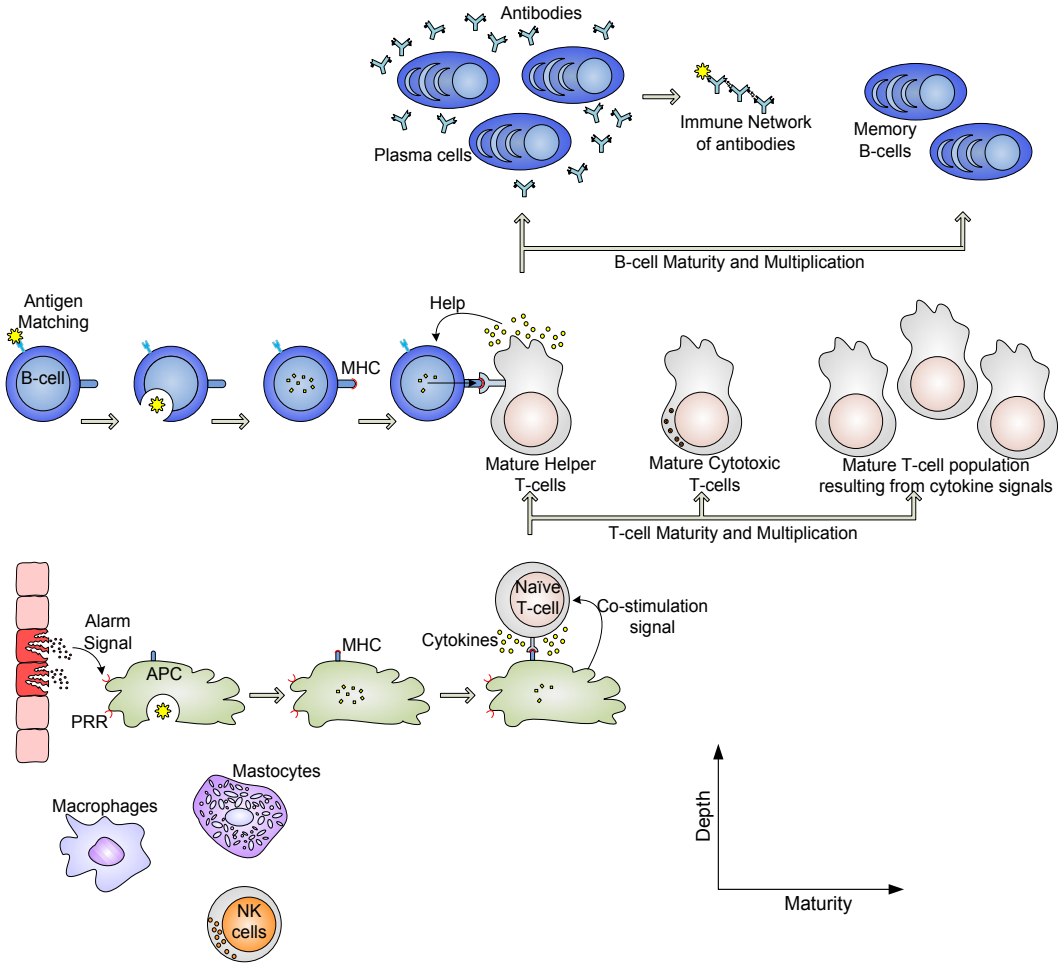


Figure 2.1: An overview of biological immune system (BIS). In BIS, different cells and mechanisms generate a defensive stance against foreign pathogens. Key components are shown for innate and adaptive immunity mechanisms.

An immune system can be defined as a collection of numerous cells and mechanisms that protect their host from infectious agents by regulating

their populations, mutual interactions and chemical secretions in response to invading pathogens. Figure 2.1, constructed from various sources [1, 37, 53], provides an overview of BIS. A number of immunological models has been suggested over the years to interpret its working. Modern immunological definitions stem from self/non-self (SNS) model resulting from Burnet's clonal selection theory [13] and Medawar's acquired tolerance experiments [8]. It was later augmented to include helper T-cells [11] and antigen presenting cells (APC) [44]. The resulting SNS model is a two signal approach that uses co-stimulation signal between APCs and T-cells and a help signal between T-cells and B-cells. Idiotypic network theory [38] explained the working of antibody-network that enables antibodies to recognize each other in addition to antigen recognition. A recent major attempt to model biological immunity is danger theory (DT) [53] in which APCs are activated through alarm signals emanating from distressed or injured cells. This suggests a third signal in addition to the two signals in previous SNS model.

2.1.1 Innate Immunity

It has been described earlier that BIS can be subdivided into innate and adaptive immunities. The innate part of the immune system works as a first line of defense and comprises various cells that recognize and consequently respond to invading pathogens in a generic manner, but does not confer long-lasting immunity. The major function of innate immunity is to call different immune cells to the sites of the infection once a non-self is identi-

fied, present antigens to the adaptive part of the immune system and remove general pathogens and dead cells. All of these functions are done through specialized cells that use different physical and chemical factors. These cells include, but are not limited to, phagocytes, mastocytes and natural killer (NK) cells [1].

Phagocytes can *eat* pathogens and have the ability to discriminate between self and non-self. These include macrophages, neutrophils and dendritic cells. These also secrete chemokines and cytokines that act together to call more phagocytic cells to the sites of infection. Furthermore, cytokines released by tissue phagocytic cells, among other functions, cause fever/inflammation and mobilization of APCs. Recognition of viral pathogens leads to interferon-production that, in turn, inhibits further replication of viruses and activates NK cells [60].

2.1.2 Adaptive Immunity

Clonal selection (CS) theory [13] is at the heart of adaptive immunity and explains how B and T lymphocytes improve their response to presented-antigens in order to acquire immunity through affinity maturation. Selection is inspired by the antigen-antibody-affinity. It states that B-cells divide when an affinity is present between stimulating antigen's epitope and B-cell receptors. These cells then mature into plasma cells and secrete antibodies. Antibodies with higher affinities are then reproduced through somatic hypermutation of B-cells. Paratopes on antibodies and epitopes on antigens work as key-lock

mechanism (complement cascade) to help other cells eliminate pathogens. The immune system retains some matching B-cells as memory cells. Moreover, it adapts by building up concentrations of B-cells as well as maintains a diversity in mutating these cells in the bone marrow.

Clonal selection theory does not explain the working of BIS in absence of invading pathogens or suppression of certain immune functions. Jerne's idiotypic-network (IN) theory [38], also known as immune network, proposes a possible explanation. It suggests that an antibody possesses a unique idiotope, similar to epitope, so that other antibodies can recognize it. The group of antibodies that share common idiotope belongs to one idiotype. This theory also states that once an antibody's idiotope is recognized by paratopes of other antibodies, it is suppressed. Consequently, antibody concentration is reduced. Similarly, once an antibody's paratope recognizes idiotopes of other antibodies or epitopes of antigens, it is stimulated. Antibody concentration is increased as a result of this stimulation. In other words, this theory tries to explain the communication between antibodies via a collective dynamic network of stimulative and suppressive interactions, suggesting a continuous communication even in absence of antigens. This is in contrast to the $antibody - antigen - only$ interactions of CS-theory. It is caused by the notion that cells within an immune system can recognize each other, in addition to recognizing antigens. This theory has been applied to a number of different applications ranging from internet security to mobile robotic systems.

Danger theory (DT) [53] attempts to explain the workings of biological

immunity in a single framework. It describes that a co-stimulatory signal from dendritic cells activates the T-helper cells. These dendritic cells of the immune system, a type of APCs, are themselves activated by *danger signals* emitted by the injured/stressed cells. Once activated, they provide a co-stimulatory signal to exhibit innate/adaptive immune response. Furthermore, dendritic cells can be immature, semi-mature and mature. Immature dendritic cells collect antigens along with safe and danger signals from its local environment like pathogen associated molecular pattern signals (PAMPS) and inflammatory cytokines. If environment is safe, the dendritic cell becomes semi-mature and upon presenting antigen to T-cells it causes the T-cell-tolerance. On the other hand, if environment is dangerous, it becomes mature and causes T-cell-reactivity [54]. Figure 2.1 illustrates the three signal approach of danger model, among other details of BIS.

2.2 Computational Interpretation of Immunity

The goal of immuno-inspired research is to translate biological immune functions into computational models to solve different problems. Although there are various computational models of immunity–mostly application specific– the following discussion presents generic models. In the CS-approach, it is important to mathematically interpret affinity between antigen and antibody. Selection, ordering and subsequent reselection of antibodies or mutated antibodies is solely done on the basis of affinity scores. It acts similar to fitness function in genetic algorithms (GA). Affinity functions are

application-specific but generally can take the following form [15].

$$A_{f_i} = \frac{A}{\sum_{i=1}^{N} (d_i + c\beta_i)} \tag{2.1}$$

Where d_i is distance between the presented antigen and the selected antibody and β_i can be defined in terms of available auxiliary data. Commonly, the distance is translated in terms of Euclidean or Hamming distances based on real or binary representations, respectively. Clonal selection is adaptive and works on the principle of antibody evolution through somatic hypermutation. The results of affinity computations, using Eq. 2.1, are sorted in ascending order which is followed by reselection on the basis of best-population-size and subsequent maturation using Eq. (2.2). Each antibody in selected-and-ordered-best-population is then cloned as described in Eq. 2.3. These clones are projected within the solution bounds. Affinities are computed again and the resulting best clones are selected. Selected best-clones then replace the antibodies in initial antibody matrix.

$$\mu_i = K_1 e^{-K_2 \cdot A_{f_i}} \tag{2.2}$$

$$C_i = A_i + \gamma \left[\mu_i \cdot \text{rand}(.)\right] \tag{2.3}$$

where, μ_i is the antibody maturation rate, K_1 is the maturation constant and K_2 is the maturation decay factor. Whereas, C_i and A_{f_i} are, respectively, the number of clones and the affinity of ith selected antibody. The scaling factor for the random number generator of the cloning expression is denoted by γ. There can be other variants of maturation and cloning expressions (Eq. 2.2 &

2.3), on the basis of corresponding representation schema. For a comprehensive computational detail on clonal selection, the reader is referred to White and Garrett [90]. Garrett [26] also presented an alternative representation to combine several B-cell representations in an attempt to combine the clonal selection and immune network approaches in a generic network.

Jerne's idiotypic network theory [38] was translated into a computational model by Farmer et al. [24] proposing a differential equation for antibody concentration A_i evolution as a function of all the stimulatory and suppressive effects as well as the natural death rate.

$$\dot{A}_i = \left[\alpha_a \sum_{j=1}^{N_a} m_{ij} a_j - \alpha_s \sum_{j=1}^{N_a} m_{ji} a_j + \sum_{k=1}^{N_g} n_{ik} y_k - \lambda_i \right] a_i \tag{2.4}$$

$$a_i = \sigma(A_i) = \frac{1}{\tau_i + exp(-A_i)}, \quad \forall\ i = 1, \ldots, N_a. \tag{2.5}$$

The equations are defined for N_a antibodies and N_g antigens. The first sum in Eq. (2.4) represents the stimulation of antibody A_i in response to the other antibodies A_j (idiotope-paratope connection). It is termed as *stimulus*$_1$ in subsequent sections. The second sum represents *suppression* of antibody A_i in response to all other antibodies (paratope-idiotope connection). The third sum models the stimulation of antibody A_i in response to all antigens (paratope-epitope connection) and is termed as *stimulus*$_2$. The last term in the expression represents the antibody death rate. The resulting antibody concentration rate depends on the collisions between antibody A_i and antibody A_j that is

proportional to $a_i a_j$. Eq. (2.4) uses a squashing function, $\sigma(\cdot)$, to control the size of a_i.

Computational interpretations and consequent algorithms, inspired by danger theory (DT), are still in their infancy. There are two interpretations following the introductory work of Aickelin et al. [2]; one is dendritic cell algorithm (DCA) by Greensmith et al. [29] while the other is toll like receptor algorithm (TLR) by Twycross [77]. Both use different aspects of DT. DCA introduces the underlying translations of PAMPS, safe and danger signals resulting from maturity of dendritic cells with the help of co-stimulatory molecule (CSM). These signals are buffered as well as the antigen. DCA, on the basis of dendritic cell maturity and migration threshold, sets the cell context. Equation 2.6, in one possible configuration of output, thus contextualizes the environment which then arbitrates the immune responses.

$$O_p = \beta \left[W_P \sum_{i=1}^{3} P_i + W_D \sum_{i=1}^{3} D_i + W_S \sum_{i=1}^{3} S_i \right], \quad \forall\ p \tag{2.6}$$

The output (O_p), for each dendritic cell in population (p), depends on pathogen associated molecular patterns (P_i), danger associated molecular patterns (D_i) and safe signals (S_i) as well as on weights (W_X) of each signal-type ($X = \{P, D, S\}$).

AIS can have different mathematical abstractions than those mentioned above. The main theme, however, would remain the same. The details of affinity functions, hyper-mutations, idiotypic stimulations/suppressions, antibody

selections and meta-dynamics are generally done on the basis of choice of the problem and associated representation schema.

2.3 Review of Immuno-inspired Robotic Applications

In the literature, AIS-based robotic applications tend to simulate robot control around small, artificial environments, generally addressing the problems of behavior arbitration and autonomous navigation. These environments are generally programmed as fixed and depicted as arenas where robots have to perform. Subsequent discussion categorizes the reported immuno-inspired robotic applications in four categories according to the underlying BIS definitions as shown in fig. 2.2. The aim of this review is to identify the underlying immunological details and associated pros and cons.

2.3.1 Robotic Applications using Clonal Selection

A small number of robotic applications are reported in the literature that are inspired by the clonal selection theory only. There exist different variants of CS-algorithms. The variations are mainly due to definitions of antigen-antibody strings, affinity computations and auxiliary functions. The underlying details of these CS-based robotic applications are tabulated in table A.1. Computational efficiencies are important but mainly depend on the string lengths and population size of mutated antibodies during the process.

Hu [30] implemented a global path planning approach using CS. Antibodies are defined as a set of nodes that represent line segments from starting

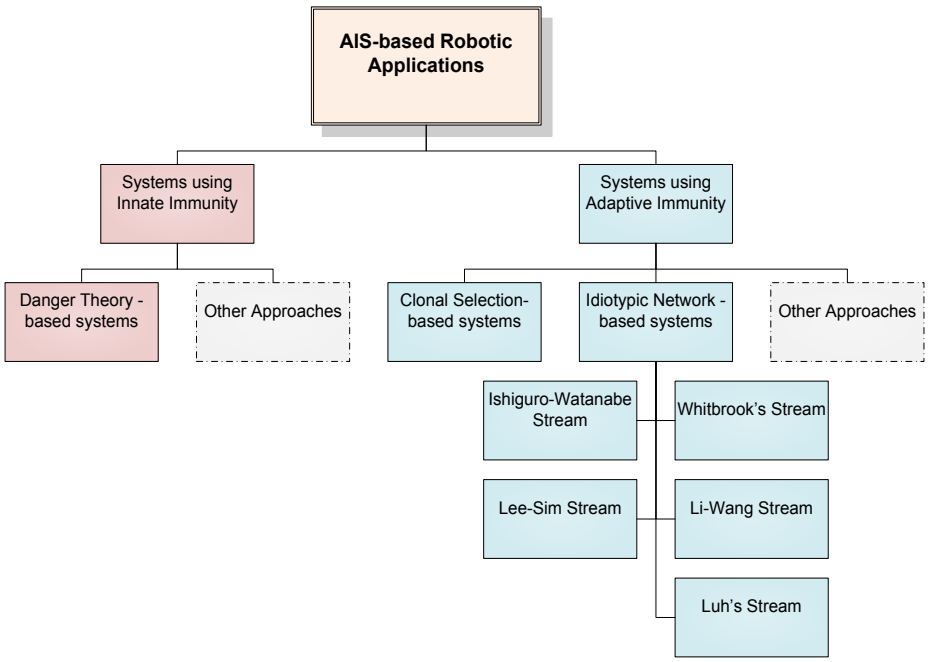


Figure 2.2: Genealogical Chart of AIS-based Robotic Applications.

point to end point. Fitness function (affinity) is defined in terms of Euclidean distance (d_i) and obstacle information (β_i) in the arena. Paths that intersect obstacles reduce the fitness value and consequent clonal selection finds the nodes towards the final destination. Global path planning has an inherent limitation of requiring a-priori information on the environment. Robot navigation, resultantly, has gone through a paradigm shift and uses variants of reactive or hybrid approaches which perform better in unknown or partially known environments. Hu's implementation also lacks antigen representation; a must for any AIS abstraction.

Wang & Hirsbrunner [83] developed an immune-mechanism-based evo-

lution algorithm (IMEA) in an off-line robot navigation task, in an attempt to avoid premature convergence during navigation, and compared it with a genetic algorithm approach. IMEA combines the concepts of genetic algorithms (crossover and mutation) with those of clonal selection (memory "updation" and selection). It is noted that this approach uses the concept of vitality (ν_i) to compute fitness function and is based on the least mean squared error between two selected paths. IMEA does not use the concepts of somatic hypermutation and cloning to evolve the solution but relies on the concepts of GA, instead.

Li et el. [48] presented a CS-inspired approach for concurrent mapping and localization in order to search the space for possible robot maps. This approach does not use the metaphors of antigen or antibody but chooses chromosomes to represent change in distance and orientation as in GA. It then uses CS for mutation purpose only. In its fitness function, described in table A.1 (in page 141), w_1 and w_2 are real numbers in the range of $(0, 1)$ and `if` $(O_{ij} > 0.5)$ `then` $\{\delta_{ij} = 1\}$ `else` $\{\delta_{ij} = 0\}$; and `if` $(E_{ij} > 0.5)$ `then` $\{\zeta_{ij} = 1\}$ `else` $\{\zeta_{ij} = 0\}$.

Wang & Hirsbrunner and Li et al. both have used GA-based crossover operators in their applications. In nature, however, BIS does not use crossover which raises a question on validity of such implementations. It is worth noting that the CS-algorithm and its variants are equally effective, if not better, in optimization tasks (e.g. [90]). A framework for establishing convergence of immunity-based algorithms is also presented in [18] and a comparative analysis is presented in [17], for various test functions. The representation of antigen

and antibody is also not explicitly defined and justified in Li's work [48]. The benefit of using a robotic application lies in its embodiment and it should, therefore, be reflected in corresponding representation schema.

The only application involving multi-robot system using CS is that of Hur, J. [33]. It performs a bomb disposal task, using CS-based antibody evolution scheme to update a lookup table (memory) that lists solutions corresponding to different states. There are some robotic applications that use clonal selection as an auxiliary function or as a metaphor only, like that of Chingtham and Nair [16]. Jun et al. [39] also used CS metaphors to augment immune network, only to transfer strategy between individual robots.

2.3.2 Robotic Applications using Idiotypic Network

The Idiotypic network (IN) has been widely used as an AIS-based approach in robotics, more than any other model, because it explicitly defines the interactions between antibodies and antigen and their resulting network. Following, is a comprehensive review of IN-based applications along with corresponding mathematical expressions in tables A.3 and A.5. The applications are further subdivided in terms of the corresponding parent technique as shown in fig. 2.2. It is observed that IN-based robotic algorithms first map the sensor data in terms of antigens, match the control strategy to the change in environment, perform network dynamics and as a result, handle the change. Antibodies are either evolved or generated by evaluating affinity functions. Mapping schema, affinity definitions and antibody specifications vary from

application to application. The most stimulated antibody, resulting from network dynamics, is fed to the system as an actuation signal. The question, however, is how to justify use of a particular theory (idiotypic network theory in this case) and to what extent a theory is applied.

2.3.2.1 Ishiguro-Watanabe Stream

This subcategory deals with binary representations for antigen and antibody. Resultant networks use Hamming distance as primary criterion for affinity computation. Consequently, antibodies are selected on the basis of their respective concentrations. The underlying principle of suppressions and stimulations follow Farmer's representation with minor differences. The details of the corresponding mathematical abstractions are listed in tables A.3 & A.5.

Ishiguro et al. [34], in 1995, implemented an IN-based approach on a six legged robot in order to acquire a gait. Each leg is incorporated with a local immune network (LIN) having four antibodies. Each antibody represents specific gait behaviors; namely backward, retract, forward and protract. Paratopes and idiotopes of all the antibodies are pre-assigned as to either support or transfer. This LIN is evolved using GA in which a "winner takes all" approach is used to select antibodies. Two types of antigen are incorporated: one to input the situation, the other to represent the coordination among local networks. Experimentation is limited to forward movement in which 18 iterations of GA establish a no fall situation. This work by Ishiguro

et al. is considered as a first attempt towards physical application of idiotypic network but ad-hoc antibody/antigen allocation restricts such systems to low complexity.

Ishiguro et al. [35, 36], in 1996, also proposed a decentralized behavior arbitration scheme to navigate a mobile robot to replenish energy, avoiding obstacles in an arena. It is noted that paratopes are modeled as desirable actions with preassigned definition of action. Idiotopes are modeled as identification numbers that are assigned according to the results of an adjustment mechanism (reinforcement). Antigens are pre-massaged in terms of object information, direction of object and current energy state. Experimental results, however, show limited results of an 18 antibody network that enables the robot to avoid one obstacle to reach the charging station. Antibody selection is done on a "roulette wheel" method. Moreover, the network does not make use of antibody meta-dynamics but uses an adjustment mechanism to select an idiotope-ID.

It should be noted that Jerne's idiotypic theory defines the idiotype in terms of a physical connection, like that of a key-lock, to identify each other. However, a BIS can open a number of locks with one key. Although this analogy is weak, antigen/antibody allocation in Ishiguro's initial work, however, does not incorporate this phenomenon. Moreover, this approach also avoids use of unstructured environment in simulations. Ishiguro's work was extended by Watanabe et al. [85] to include an off-line innovation function. This innovation function is based on a genetic algorithm with a "mixing pot" method for

crossover operator. It is noted that antibodies are retained as behavior modules. The initial problem was also extended to add *garbage collection* behavior in addition to existing *obstacle avoidance* and *energy replenishment* behaviors. The drawback of this approach, as well as of Ishiguro's, is the definition of antibodies as behavioral modules. This approach forces one to define behaviors ahead of time with no possibility of behavior inclusion/evolution.

Michelan and Von Zuben [56] improved Ishiguro's model by incorporating a GA-based antibody evolution mechanism. Idiotopes are modeled as a set of stimulated antibodies for the network. Antibody affinity is computed on the basis Hamming-distance evaluation. GA-based adjustment mechanism uses a 40% crossover and 1% mutation with elitist selection. Fitness function is based on the number of collected-and-transferred garbage, recharges and collisions. It should be noted, however, that a BIS has an inherent mechanism to clone antibodies using somatic hypermutation. The above mentioned models use other algorithms for similar purposes. This raises a question on the degree of AIS implementation.

Vargas et al. [80] attempted the same garbage collection application using a learning classifier system in addition to the existing platform provided by Michelan and Von Zuben. This model, named CLARINET, adds a learning classifier system to classify antigens and antibodies. Antibody structure is, however, restructured in terms of antecedent and consequent parts to represent paratopes and antibody connections to represent idiotopes. This addition makes antibody network more flexible but requires more computation effort

as classifiers are updated both before and after the immune network dynamics. They also implemented the GA-based immune network on Khepra robots [79].

Krautmacher and Dilger [43] tried to implement a simplified rescue scenario involving a single robot. Antigens are binary-coded information of object type and position. The algorithm then uses coordinate transformations for network dynamics in which no meta-dynamics is incorporated. The rest of the implementation is same as in Watanabe's approach.

Wang et al. [84] used the IN-approach of Ishiguro in conjunction with obstacle restriction method (ORM) and reinforcement learning (RL). This application is a single-robot path-planning exercise in which two types of antibodies are defined: one to represent obstacle avoidance behavior and the other for goal-seeking. Antigens are defined as a binary-coded data of obstacles/goal in terms of task proximity (near or far). An expression similar to a T-cell metaphor, as in Luh et al. [51], is also used to help suppress either of the behaviors. This also replaces the need to define $stimulus_2$. Moreover, cloning or meta-dynamics is not defined in network structure.

Tsankova et al. [75], in 2007, applied Ishiguro's network to implement stigmergy[2]-based foraging behavior. This work uses different scenarios to collect pucks with single/two robot(s), with one network for goal following behavior and one to pick and drop the pucks. The report also compares the

[2]Stigmergy is a mechanism of indirect coordination between agents or actors. The principle is that the trace left behind in the environment by an agent previous action affects/simulates the performance of future behavior of the same or other agent.

results with Braitenberg's 3C and Q-learning robots. This research does not add to Ishiguro's interpretation however, the experimentation establishes a comparative analysis with two well-known approaches.

2.3.2.2 Whitbrook's Stream

Whitbrook et al. [86] solved the maze-world problem with extensive experimentation using three approaches: RL, RL with simple idiotypic system and RL with full idiotypic AIS. The system uses 8 predefined antigens having priority levels assigned to pre-massaged data that translates sensor info into a "situation". Sixteen (16) antibodies have predefined behaviors with speed specifications. Idiotopes are fixed while paratopes are predefined that have some adjustment flexibility through reinforcement. Affinity computation is done as in Vargas et al. [80]. It is noted that antibody meta-dynamics is not implemented. Moreover, system uses a-priori information in antigen, antibody and idiotope matrices with limited adjustability of paratopes. Although results show that robot with full feedback performs better in terms of escaping traps by establishing idiotypic network, the system should be able to adjust its internal values automatically either through T-helper cells or through antibody evolution.

Whitbrook further extended her work by incorporating GA to evolve behaviors [87]. This GA supported long term learning (LTL) combined with short term learning (STL). The idiotypic immune network was tested against STL only approach in [88,89]. The underlying notion that AIS can only exhibit

short term learning can be questioned as it is dependent of the system's meta-dynamics that can be adjusted to retain memory for a longer period.

2.3.2.3 Lee-Sim Stream

Lee et al. [46] executed a swarm intelligence task involving multiple robots. Task density, either high, medium, low or nil, is represented as an antigen. Antibodies are defined as four behaviors of *aggregation*, *random search*, *dispersion* and *homing*. Sensors detect the task concentration that is then used in a fuzzy inference system (FIS) function to output a stimulus-value for Farmer's equation. The resulting concentration is then used to stimulate other robots to do the same task. Metaphors of plasma and deactivated cells are used to incorporate some level of meta-dynamics. This approach also suffers from the inherent problem of *a-priori* behavior specification. Jun et al. [39] extended this work by incorporating T-cell metaphor to represent control parameters. This adds another layer in the network and resets the antibody concentrations once an antigen is removed.

2.3.2.4 Li-Wang Stream

Li and Wang [47] implemented a sheep-and-dog problem using predefined coefficients to compute affinities. Antibody network dynamics is replaced by an algebraic expression that takes into account the usual stimulations and suppressions along with a T-cell function and a linear *death rate*. The environment is translated into antigens by tabulating positions of dog and sheep in

a matrix X. The antibody matrix Y also has previous information of actions corresponding to each entry in the antigen matrix X. Only five actions are possible. This research does not make any distinction between sheep and dog in terms of their embodiment. Moreover, the network uses a manual mechanism to perform coefficient selection that limits its adaptability.

Duan et al. [23] extended the work of Li and Wang to perform a predator-prey experiment with 2 predators and one prey, each having a small antibody network that can communicate with each other except in *pursuit domain*. Antigens are of two types: one has environment information in terms of position data and other handles communication signals. Two different antibody structures are implemented for predator and prey robots. Predator has six actions to arbitrate from while prey has three behaviors to choose from, on the basis of synthesized immune network as in Li and Wang [47].

2.3.2.5 Luh's Work

Luh has presented three different applications that use different immune metaphors. His approach is based on real data representation schemes. In 2002, he with Cheng [49] presented a food foraging application that uses APC modules to assess the environment and T-cells as a RL mechanism. T-helper-cells are used as an *adaptive critic*. Luh et al. [51], in 2006, implemented a robot soccer application using immune network. The antigen is sensor information that is mapped to have three components: one is distance between ball and goal, the second is distance between ball and robot and the third

is crowd data. Each of these components corresponds to a fuzzy function to find affinity value. Average of all three affinities, through fuzzy membership functions, is computed in terms of a 6x6 affinity matrix. A T-cell function is incorporated that acts as a reinforcement. Antibody meta-dynamics is not implemented since there is no repertoire maintained as memory. Zhang and Lu [94] reproduced this approach using four antibodies instead of six for each robot.

Luh and Liu [50], in 2008, solved the robot navigation problem using the reactive-IN approach with fused data representation. The antigens are vectors of azimuthal angle of goal, distance information of each sensor and sensor location on the robot periphery. Antibodies are defined as *steering directions* (θ_i). Stimulation and suppression due to antibody-antibody interactions is defined as cosine of the difference between antibodies. Stimulation due to antigen interaction is defined in terms of attractive/repulsive forces of *goal seeking* and *obstacle avoidance*. In order to escape robot from trapping in local minima, an adaptive virtual target method is also used. The weighing mechanism of attractive/repulsive forces is manual. Therefore, it is not clear that how a robot manages to arbitrate the two behaviors.

Dehuai et al. [22, 92] modified the work of Wang et al. [84] by defining antigens in terms of *task density* (high, low or none) and combining antibody structure in one representation. It is also noted that Farmer's equation is not solved by an ODE solver but antibody concentration rate is related directly to behavior modules. Moreover, Hamming distance is replaced with Luh's

expression of $cos(\Delta\theta)$ to define antibody stimulation and suppression. This application is also a single robot path planning exercise in which two types of antibodies are defined: one to represent obstacle avoidance behavior and the other to seek goal behavior.

2.3.2.6 Non-Farmer Approach

Mitsumoto et al. [58, 59] presented an IN-based approach to control a population of multiple robots according to assigned task of load transfer from one station to two storage docks. Each task assignment is treated as an antigen that disturbs the existing population of robots. The algorithm then reconfigures to attain new stability (homeostatic state) by sharing message-antigen with other robots. Each robot, treated as a B-cell, has predefined modules to set global states and their behavior strategy. Resultantly, the network is limited only to regulate robot population.

Sathyanath and Sahin [71] and Opp and Sahin [63] used a mine detection task to perform single objective task with a fixed number of robots using a non-idiotypic approach. Antigens are modeled as mine locations whereas antibodies are defined as robots. The communication between antibodies is a network that provides antigen-locations. Robots are, resultantly, stimulated to move toward the mines in order to defuse them. Suppression is implemented when no antigen is detected and results in random movement. This is unlike Farmer's interpretation of Jerne's idiotypic network theory that ensures communication even in absence of antigen.

Generally, it is because of this notion that cells within an idiotypic network can recognize each other, in addition to recognizing antigens. The IN-approach is applied on mobile robotic systems (MRS). Any change in environment is detected as an antigen. Possible steering directions/behavior-modules are represented as antibodies. The most stimulated antibody (resulting from immune network approach and supplementary methodologies) is fed to the system as an actuation signal.

2.3.3 Robotic Applications using Danger Theory

Danger theory is a newer definition of BIS working and, therefore, very few robotic applications are reported in this category. Dendritic cell algorithm (DCA) by Greensmith [27–29] incorporates only one aspect of danger theory. It mimics the working of antigen presenting cells within a BIS and, therefore, requires other immuno-functions to fully implement a three signal model.

Oates et al. [62] used DCA in a mobile robotic security application for classification purposes. Augmenting the *subsumption architecture*, the robotic DCA is implemented as a stand-alone behavioral module. DCA processes the sensor data as antigens and generates signals that are either *safe*, *dangerous* or *PAMP* [2]. The output of the DCA provides a base for subsuming the behavioral modules. It is to be noted that in subsumption architecture, there is a disagreement among various behavioral modules e.g. react to bumpers, recover from stall, avoid obstacles and explore. Brooks [12] suggests that this can be solved by allowing components at one level to subsume components at

a lower level. It is because of this reason the approach is called subsumption architecture. This application, however, is a classification problem that does not fully incorporate the behaviors necessary for navigation through a maze.

Prieto et al. [68] implemented a preliminary work that uses DT on a metaphorical level only and lacks the necessary mathematical interpretations. His application is a soccer goalkeeper strategy in which APCs are ID of predefined strategy whereas antigens are composed form the detection of opponent and ball in the home side. Signal one, two and three correspond to the respective closeness of the ball to the goal.

2.4 Literature Findings and Conclusions

In advent of newer definitions of BIS and current trends in robotics, it is important to categorize these applications in terms of underlying immune definitions, computational details and deficiencies. Consequently, point towards future directions in AIS research. This section initially discusses the findings on AIS-implementations and then focuses on the issues pertaining to robotics in the reported literature while presenting an inline critique. Immuno-inspired applications were reviewed in light of their possible application in a heterogeneous mobile robotic system. The major questions for each AIS are:

① Is it *flexible* to include various morphologies of mobile robots?

② Is it *scalable* to add more robots in different roles?

③ Is it *adaptive* to evolve suitable behaviors?

④ Is it able to *communicate* strategies across the board?

2.4.1 On using AIS

It is important to indicate that some aspects of BIS are still being investigated. The CS theory, the oldest of the considered theories, is focused on one signal approach in which antigens binds with receptors on a B-cell. The danger theory is the newest in self/non-self models of BIS and follows a three-signal approach, as shown in fig. 2.1. Therefore, year of publication is relevant to identify the validity of corresponding BIS explanation, used in the application. Moreover, there exists a researcher's dilemma that hints to 'what to take' and 'what to leave' in order to solve a particular problem, especially when a set of theories explain different aspects of a complex phenomenon. It is of no use that the whole BIS is replicated to solve a relatively simple problem but, at the same time, one should not fall victim to a single aspect of BIS as well.

It is vital to identify the steps through which any immunity-based application should be designed. Castro [15] defined a layered framework, to serve as a touchstone, in which an AIS undergoes the processes of *representation*, *affinity measures* and *immune algorithms* to solve a problem in an application domain, as shown in fig. 2.3. An additional step to define immune-entities is augmented in Castro's original framework because a combined *innate-and-adaptive immunity* is envisioned in this research, contrary to Castro's *B-cell-only* approach. This step should facilitate the designer to choose from multiple *lymphocytes* along with their specialities and interconnections. The following discussion is structured along the lines of Castro's framework.

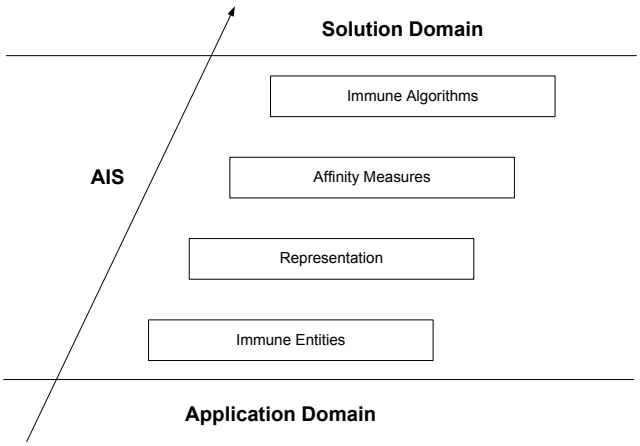


Figure 2.3: AIS-based framework for problem solving, adapted from Castro [15].

The *representation schema* refers to the type of data structure used in defining signals, antigens and antibodies of an AIS. It is important because it dictates the subsequent *affinity computations* (e.g. Hamming for binary and Euclidean for real representation) as well as candidate *immunity algorithms*. Immunity-based robotic applications indicate that the Ishiguro-Watanabe stream uses binary representation, whereas others use variants of real representation. The major problem with the representation scheme used in the Ishiguro-Watanabe stream is the constraint of pre-programming all the possible behavior modules ahead of time and linking them with binary representations. Ideally, intelligence should emerge irrespective of the hardware configuration of the robots. Although, this approach is able to communicate strategies because of idiotypic connections and, is adaptive to evolve behav-

iors if an evolution strategy is implemented, it is inflexible to handle robot morphologies because a fixed-length antibody structure is implemented with no provision to relate the robot-embodiment to the network.

The applications using real representations, on the other hand, use different methods to compute affinities e.g. Whitbrook et al. [86] uses strength of match approach, Luh and Liu [50] use a combined stimulative & suppressive trigonometric expression, whereas Li and Wang [47] incorporate a string matching function. Although, it is difficult to summarize the ad-hoc manner in which applications are reported in this category, it can be concluded that a network-alone approach, like that of Luh and Liu [50], reduces robotic systems to be *reactive-only* that have no evolution resulting from cloning & hyper-mutation. Moreover, these applications are adaptive and can also communicate antibodies but are inflexible to include various robot morphologies in the network and require pre-assigned behavior specifications.

Although, *Ishiguro-Watanabe*'s mode of representation has a problem that it's network does not have any information of the robot body but at the same time has a more generic structure. *Luh*'s approach, however, makes use of fused data representation that includes some information of location of sensors within the network. The *selection of representation schema* in AIS-based robotic applications is, therefore, tricky in terms of whether we should include the robot embodiment information in the network with it's consequent loss of generality. An extensively detailed account of representation schema of all the reported immuno-inspired robotic applications is presented in tables

A.1 and A.3.

The next step along the lines of Castro's framework is to define *affinity measures*. Although, Hamming or Euclidean distances are the basic choices for binary or real representations, respectively, these do not necessarily meet all the requirements for affinity computations. Affinity expression in clonal selection theory (eq. 2.1) indicates provision of an auxiliary data (β_i) which is scaled through c. The selection of this auxiliary data is application-specific and the final affinity expression may not take the form of eq. 2.1, as indicated in table A.1.

The last step in designing an AIS-based application is to *encode an immune-algorithm*. It can be inspired from a spectrum of immune functions, ranging from innate to adaptive components. It is observed that innate immunity has largely been ignored because only recently the innate immunological functions has been sufficiently explained to be encoded. CS and IN theories are applicable to adaptive components of immunity and therefore should carefully be used in conjunction with *holistic* definitions e.g. *danger theory*.

The major criticism on Jerne's idiotypic network theory is in relation to the size of its network [45]. This criticism arises from the argument of how can every antibody recognize every other antibody in a possible network of millions of cells (with current estimates of more than 10^{12} lymphocytes). Similarly, the network structure in terms of its symmetry has its own share of arguments [20]. These arguments are not countered in the observed robotic applications. It can, however, be argued that such applications do not incorporate a network

of millions of antibodies. In case of behavior arbitration, the network only requires a handful of antibodies. In the case of multi-robot applications where robots are generally modeled as B-cells, the size of the network can not exceed a certain limit because of inherent limitations of cost and size of the arena.

Apart from the conventional approach of using an ODE solvers for antibody concentrations in Farmer's expression (eq. 2.4), a discretized version using a bilinear transform is employed in Krautmacher & Dilger [43] and an algebraic equivalent, named as *synthesized immune network*, is used in *Li-Wang stream*. The choice of a particular implementation strategy or a solver is normally dictated by computational requirements of an application and available computational resources. Recently, theoretical issues relating to different AIS algorithms are being raised. The small size of immune network in case of robotic applications induces an effect of discreteness, resulting in difficulties to analyze them using standard techniques [74].

Literature indicates that some auxiliary functions or subsystems are also required to support the core immune-algorithm, like *reinforcement learning* (RL), *fuzzy systems* (FS) and/or *genetic algorithms* (GA). It would be more appropriate to use a computational equivalent from BIS, if available. For example, nature uses somatic hypermutation to evolve antibodies but some researchers have used GA instead (e.g. [56,80]). It would have been logical to use what nature has chosen for a particular purpose. Similarly, some researchers have used RL (e.g. [35, 84]) when nature uses similar approach of T-helper-cells. This identifies that the *degree of biological inspiration* can be deeper than

some applications show it to be. Moreover, there should be investigations to establish that BIS-inspired auxiliary function(s) can be as effective as their alternatives.

Antibody death is also an important factor in IN-approaches. A constant antibody death rate, as employed by all the applications, does not serve a purpose when no memory is maintained or when antibody meta-dynamics is skipped in its implementation. Maintenance of successful B and T lymphocytes should gradually decrease the computational load on respective AIS.

Danger theory can be considered as an extension of self/non-self models. DCA is based on one aspect of this theory that relates APCs to their maturity on the basis of danger/stressed signals in the system. Resultantly, DCA limits itself to the initiation of immuno-responses because the theory itself puts limits on that. It can be used in behavior arbitration on the basis of environment contextualization but then it should be supplemented with B-Cell and T-Cell algorithms to complete a three signal immune function [2]. Only one application is reported in literature that uses DCA in a robotic application. There is a lot of "potential" in terms of using innate immunity in conjunction with its adaptive counterpart. As a starting suggestion, conflicting objectives during robot navigation can be tested with DCA which is currently limited to static data. Moreover, fuzzy weighing in some instances can be replaced with a DCA to co-stimulate different behavioral modules. The role of T-cells in helping B-cells can also be further refined to a level of developing adaptive critics as well.

It is also observed that important immune-functions of *chemotaxis*, *monocytic movement*, *phagocytosis*, *inflammation* and *antigenic presentation* are not employed in immuno-inspired robotic applications. The monocytic activity is important because monocytes walk a biased random walk to the gradient of chemoattractant/chemorepellent environments. This phenomenon can be used to *move* robots. The function of phagocytosis is central to innate immunity because macrophages, mastocytes, dendritic cells, etc. are all phagocytes that *eat* the bacteria. Similarly, internal monitoring of homeostasis is possible through inflammation module.

2.4.2 On immuno-inspired robotics

Current trends in robotics, on the other hand, have upgraded from reactive paradigm to hybrid and probabilistic robotics in order to counter uncertainties in sensing and modeling [73]. Moreover, single robot applications have gathered more robots to implement swarm intelligence [70]. Heterogeneous mobile robotic systems, a new trend, involves using robots of different capabilities performing jobs in unstructured environment. Most of the reported immuno inspired applications, however, involve either single robot or multiple robots of same type. On the other hand, heterogeneity in multi-robot systems requires a generalized representation scheme that can handle robots of varying capabilities in terms of their sensory and actuatory capabilities. Moreover, the conventional idiotypic network approach couples antibodies with predefined actions/behavior-modules that, in turn, limits inclusion of

new behaviors or behavior evolution in a heterogeneous robotic system.

Robot trapping in a local minima is the most common drawback of using a reactive approach. In classical mobile robotics, many trap-escaping schemes have been tried and investigated. Potential field method [32, 52], numerical potential field method [7], virtual target method [95], virtual force field method [9], vector field method [10] are some of the methods that are used to help local minima recovery in robot navigation scenarios. Immuno-inspired methods are also being designed to handle the issue because of adaptive nature of these algorithms e.g. virtual target method by Luh and Liu [50].

Most of reported literature uses simulations and do not implement the algorithms on real systems. It should be noted that there are issues pertaining to non-holonomic nature of most robotic platforms. In simulations, it is much easier to implement a robot as a dot, irrespective of its dynamics. Similarly, the detection of obstacles, walls or targets is difficult and pose a lot more implementation issues.

It is concluded that a deeper biological inspiration is required because a single aspect of AIS may not be sufficient to incorporate a successful robotic system. Auxiliary functions should be taken from their computational equivalents within BIS, where available. An *all-encompassing* AIS has functions that provide a distributed network structure like idiotypic network, reinforcement learning like T-cell algorithms, evolutionary mechanism like somatic hypermutation, short term learning like meta-dynamics and weighted sum of attractive/repulsive forces like dendritic cell algorithm. A two layered approach can

be one of the solutions where one layer corresponds to antigenic data and the other to environment contextualization in terms of safe or dangerous signals.

A network-alone approach reduces robotic system to a reactive one that has no evolution resulting from cloning & hypermutation. Current trends, on the other hand, are more inclined towards behavior evolution rather than behavior arbitration. It is also concluded that with a deeper BIS inspiration it is possible to add stochastic nature of clonal selection to the deterministic approach of idiotypic network. The benefit of using robots as an application is in its embodiment. Fear of unknown environment can be reduced by knowing something about robot. The information of sensor-locations and system-dynamics can, therefore, be a part of representation schema.

Selection of a particular robotic application is also important. *Search and rescue* scenario involving heterogeneous robots offers a comprehensive application that uses different robot configurations to accomplish a wide variety of tasks, ranging from single robot navigation through obstacles to multi-robot coordinated navigation in rescue. Robot taxonomy is important aspect to be considered as well, especially when one wishes to develop a general algorithm for a number of robot platforms. It is also identified that benchmark problems should also be tested to validate an algorithm.

2.5 Relationship between Dissertation Research and Prior Work

Figure 2.4 shows the immunological functions that have been used by different research groups. The indicated approaches do not show the auxiliary functions that have been employed outside the realm of immunology; a detailed account, however, is presented in tables A.3 & A.5. It is evident that a holistic approach, combining innate and adaptive functionalities, is open for investigation. The dissertation research is, therefore, an attempt to formulate a *combined* framework of innate and adaptive immunities for an HMRS.

The structure of different functionalities within IBF are distributed but interconnected. Therefore, the representation is needed to be compatible across the different immunological levels. It means that if a 16-bit binary string is used to define an antibody, the contextualization through dendritic cells should also be compatible with the 16-bit binary string.

2.5.1 Computational cost of antibody concentrations

The most commonly used immunological interpretation is that of idiotypic network. Farmer's expression (eq. 2.4) shows the differential equation corresponding to the antibody concentrations of idiotypic network. The computational effort required to solve the equation depends heavily on the number of underlying antibodies. Since most of the robotic applications use single robot to exhibit network's power, its computational cost is ignored. But it is necessary to evaluate it for a multi-robot application. A test function was

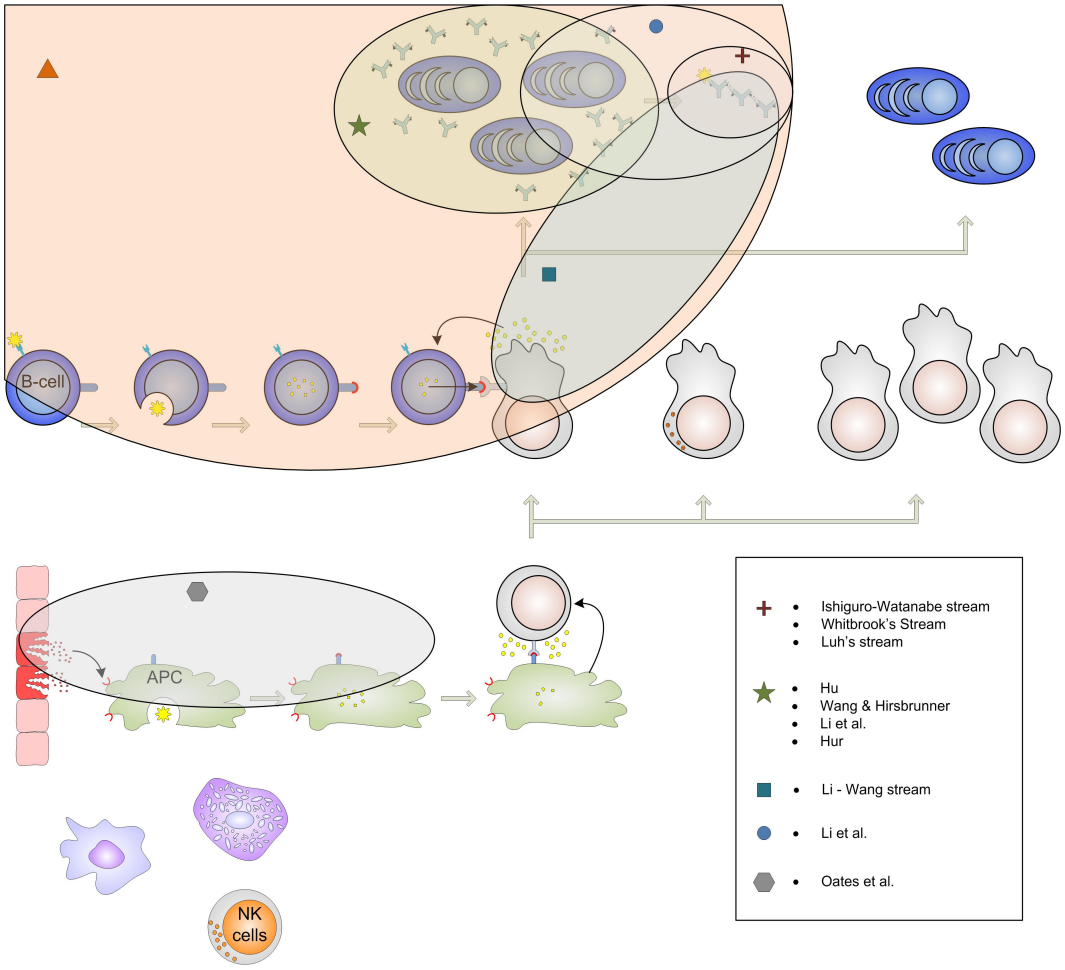


Figure 2.4: A review of immuno-inspired robotic applications

designed to find the computational costs of an idiotypic network without the incorporation of clonal selection. Figure 2.5 illustrates its results with respect to a gradual increase in the number of antibodies (A_i). The solvers used for the function are: Runge-Kutta 4th order ODE solver with 100 fixed steps and MATLAB's standard *ode45* solver. The simulation was executed using MAT-

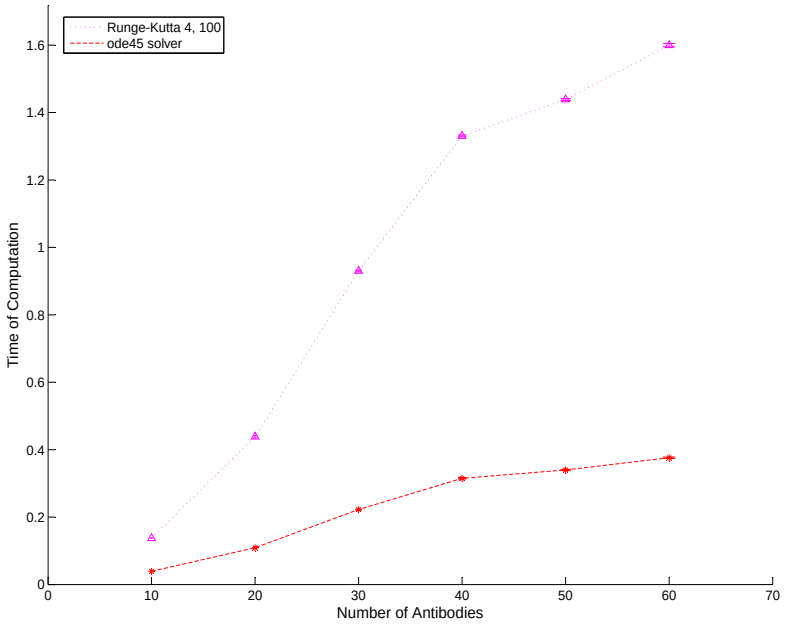


Figure 2.5: Time of computation with respect to increasing number of antibodies (A_i) in a idiotypic-network-only simulation

LAB, on an Intel ® Core™ i7-2675QM CPU @2.20GHz with 6.00GB RAM.

The results indicate that by increasing the number of robots in the arena, the computational cost rises because of the increase in number of antibodies. The dissertation research, in this context, offers an alternative by employing a population based innate immunity module that is computationally *friendly*. The detailed comparison is presented in Chapter 6. Figure 2.5 is included to signify a limitation of using a conventional approach of idiotypic network in an HMRS.

2.5.2 Robot-Heterogeneity

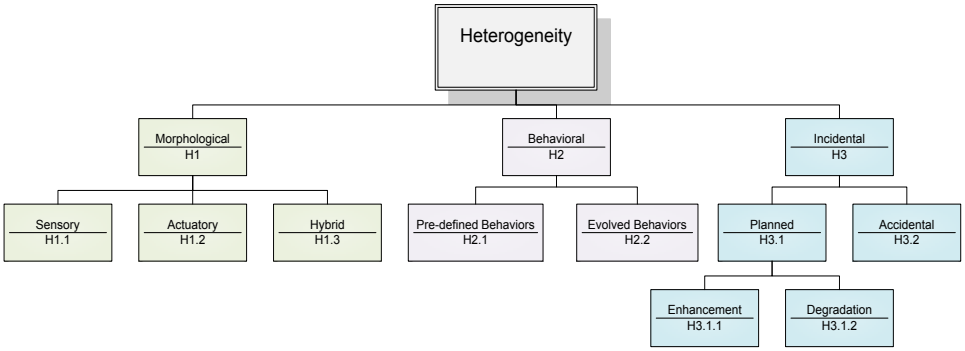


Figure 2.6: Genealogy of robot's heterogeneity

The heterogeneity in mobile robotic applications can be subdivided into three categories: morphological, behavioral and incidental. Morphological heterogeneity [65] refers to the differences among hardware configurations of robots. Figure 2.6 further classifies it in terms of sensor types and locations (H1.1), actuator types and locations (H1.2) and hybrid (H1.3). Behavioral heterogeneity [4] refers to a diversity in robots' behaviors in multi-robot scenarios. It can either be a consequence of pre-programmed behavioral diversity (H2.1) or a result of underlying behavioral evolution (H2.2). The third category is incidental heterogeneity. An upgrade in robot's existing hardware can be an example of *enhancement* (H3.1.1). Alternatively, a robot may decide to turn some of its *specialty* sensors *ON*, in certain scenarios. Similarly, a degradation (H3.1.2) can also be planned. Moreover, accidental heterogeneity (H3.2) can result from an un-planned accident during a robot's activity.

The dissertation research uses the classification of robot-heterogeneity

to test the IBF. The nomenclature used here is repeatedly referenced in subsequent discussions.

2.6 Chapter Summary

A concise introduction to the relevant concepts of immunity is presented in the start. It is followed by an exhaustive review of the literature on immunity-based robotic applications. It is found that the concepts of innate immunity are required to be included in robotic applications. Moreover, multi-robot applications, in general, and heterogeneous robotic applications, in particular, are required to be tested using immunological functions, to illustrate the pros and cons of using AIS. Different voids in terms of pre-defined antigen/antibody specifications, determinism in IN-approach, size-limitations of the network and irrelevance in auxiliary functions are also indicated. Although, pointers towards objectives of IBF and its development are given during the critique of literature, a relationships between dissertation research and prior work is also established in the last section.

Chapter 3

Innate Component of the Framework

3.1 Inspiration from Innate Immunity

A robot moves in an arena, unaware of what is out of sight, while reacting probabilistically to the sensed-data is the crux of probabilistic robotics. Similarly, a monocyte moves in a body, unaware of what is not sensed, and walking a biased random walk towards a sensed-vicinity is the crux of phagocytosis. A robot taking a leaf out of a monocyte's book is, therefore, expected to navigate successfully. An inspiration from innate immunity offers more; the monocytes secrete cytokines that attract/repel other cells. Therefore, a path taken by the first robot can attract other robots to flock together or repel each other to disperse in the arena. Cytokines also diffuse with time. The robotic system, therefore, can keep a short term memory of previous experiences. But what if the recent experience was bad. Innate immunity functions can offer their robotic brethren benefits from inflammation and dendritic cells' maturity to trigger adaptive immunity, in such a scenario. It is appropriate now to define the metaphors used in designing an HMRS (Heterogenous Mobile Robotic system) using innate immunity. Following table 3.1 presents the details.

Figure 3.1 shows the details of a robot with innate component of im-

Table 3.1: Metaphors for Innate Component of Immunity-based HMRS

Innate Immune System	**Robotic System**
Bacteria	Environment around a robot through instantaneous sensory information
Monocyte	Steering agent (hosted along the robot's periphery)
Chemotaxis	Movement of steering agents within a robot
Chemoattractants	Mapped sensed & stored data, a robot is attracted toward
Chemorepellents	Mapped sensed & stored data, a robot is repelled from
Cytokines	Previous traces of robot's agents, diffused over time
Biased random walk	Movement of each steering agents according to presented bias
Inflammation	A combined effect in a window of recent experiences
Dendritic cell	Transition agent to arbitrate immunity levels
Danger signal	Mapped inflammation or external danger (to jump the immunity level)
Safe signal	Mapped inflammation (to remain in current immunity level)
PAMP signal	Mapped data (to react to peripheral sensory information)

munity. The robot can communicate with other robots if its innate immunity fails to maintain a low inflammation level. Similarly, other robots in the arena can also communicate with this robot through antibodies. It is noteworthy that antibodies are resulted by triggering the adaptive immunity of the system and, therefore, hold a higher priority. If the antibodies are communicated back from the AIS, the robot will act like a *Yuri's slave*[1]. Other robots in the arena may have different roles and morphologies, as indicated in fig. 1.1. Subsequent sections discuss all the mathematical abstractions and relevant

[1]Command & Conquer: Red Alert 2, by Westwood Studios

implementation details.

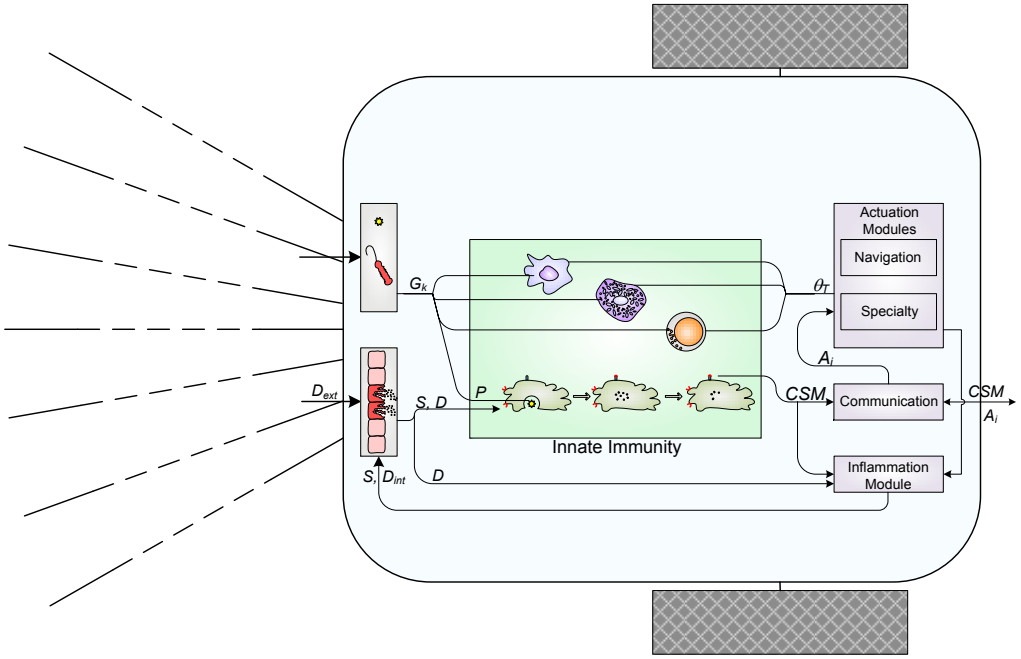


Figure 3.1: A generalized structure of innate immunity in a robot

3.2 Mathematical Abstraction

Just as GAs (Genetic Algorithms) need to encode solutions (*phenotypes*) into chromosomes (*genotypes*), the major research task of applying AIS to our engineering problem is to translate a biological phenomenon into a robotic system. Furthermore, it is not desired to replicate all the biological details. Keeping these guidelines in mind, there are two options to model the cell behavior in innate immune system. One is to use a deterministic model of a cellular activity and the other is to opt for a stochastic model. It is im-

portant for a predator to *wander* first; equally important for an antelope is to *forage*. In an SAR (Search And Rescue) scenario, it is useful for a search robot to look *here-and-there* for possible targets to rescue. These behaviors favor a stochastic approach because of the degree of inherent uncertainty, associated with the environment [73]. This research, therefore, uses the stochastic modeling approach for its innate immunity module.

The following subsections provide mathematical details of the innate component of the framework, including *representation*, *phagocytosis*, *inflammation*, *dendritic cell activity* and *initiation of T-cell maturity*.

3.2.1 Representation

The first step is to represent bacteria. Figure 3.2 illustrates the methodology to represent bacteria as mapped sensory data in the vicinity of the robot. The monocytes are housed within the robot, around its periphery, and are shown in green. In the next chapter, the same methodology is applied to define antigenic data for the adaptive component of immunity. The robot configuration, $\Big\{[-\pi/3, \pi/3],\ 16,\ \pi/24\Big\}$, indicates that 16 sensors cover a frontal azimuth of $2\pi/3$ with $\pi/24$ intervals.

The *bacteria* is defined as a matrix ($\mathbf{B}$) of individual sensor readings ($\mathbf{b}_n$). Each bacterium in matrix $\mathbf{B}$ is defined as a location in cartesian coordinates, rotated to the azimuth of each sensor-reading. If the corresponding sensor detects either a target or an empty space, a bacterium is recorded to otherwise zero values. Equations 3.1 & 3.4 show the details of bacterial rep-

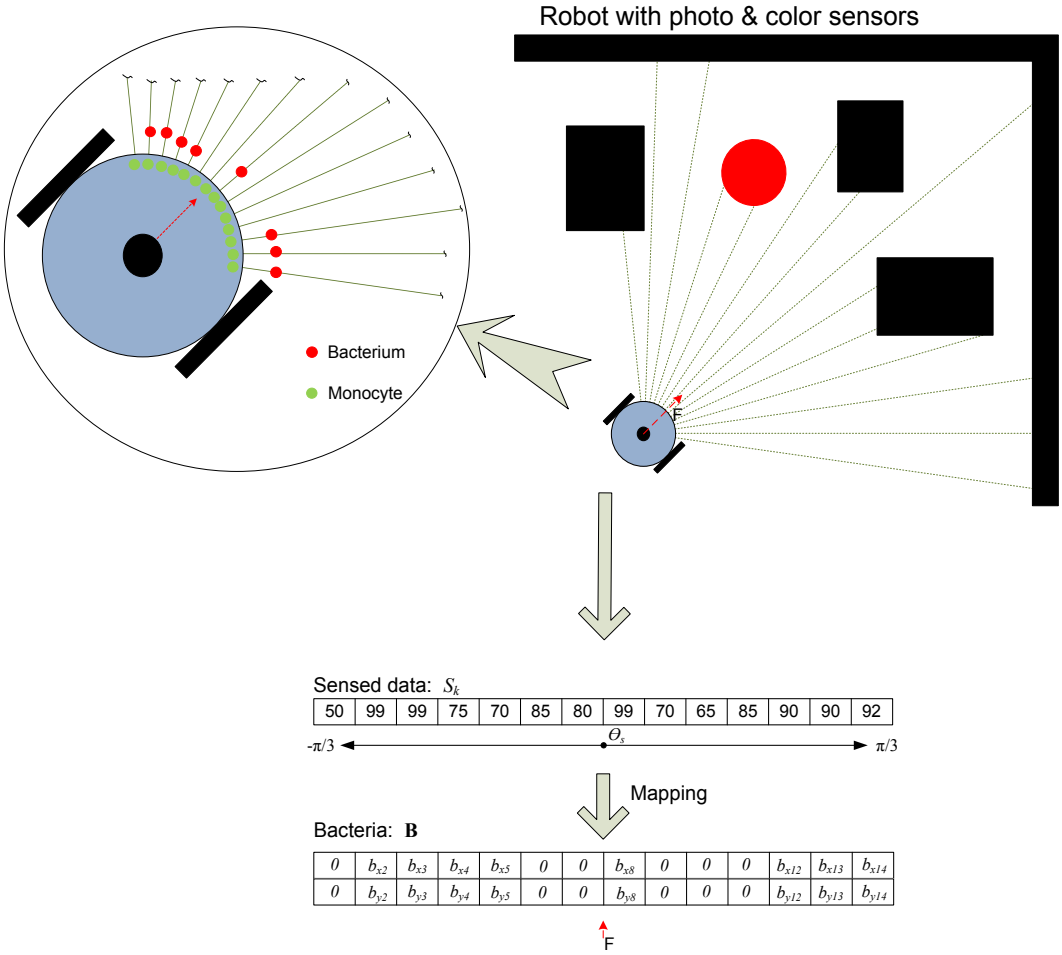


Figure 3.2: Structure of bacterial representation. The illustrated robot has a $\left\{[-\pi/3, \pi/3],\ 16,\ \pi/24\right\}$ configuration.

resentation. The total number of bacteria at each instance is equal to the

number of nonzero columns in $\mathbf{B}$.

$$\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_{N_s}] \tag{3.1}$$

$$\mathbf{M} = [\mathbf{m}_1, \ldots, \mathbf{m}_{N_s}] \tag{3.2}$$

$$\mathbf{C} = [\mathbf{c}_1, \ldots, \mathbf{c}_{N_s}] \tag{3.3}$$

In order to respond to bacterial invasion, monocytes are defined as a matrix, $\mathbf{M}$. Monocytes are cells that can develop into macrophages or dendritic cells. The macrophages move towards the site of bacterial activity by following a *biased random walk* to the secreted chemoattractants, whereas the dendritic cells go into maturity once they engulf the bacteria. It is, therefore, important to represent monocytes as steering agents. The subsequent computations move the monocytes and their consequent probability densities to navigate the robots. Equations 3.2 & 3.5 provide the details of monocytic representation.

Similarly, each cytokine is represented as a previous location of respective steering agent (monocyte), according to eq. 3.6. The matrix $\mathbf{C}$, in eq. 3.3, is constituted of individual cytokine values.

$$\mathbf{b}_n = \begin{cases} \mathbf{p} + \left(\frac{d(\theta_s)}{\alpha_b}\right) \mathbf{R}(\theta_\mathbf{s})\hat{\mathbf{u}}, & \text{if } \texttt{target} \vee \neg\texttt{obstacle}; \\ \mathbf{0}, & \text{otherwise.} \end{cases} \tag{3.4}$$

$$\mathbf{m}_n = \mathbf{p} + \mathbf{R}(\theta_\mathbf{s})\boldsymbol{\epsilon} \tag{3.5}$$

$$\mathbf{c}_n = \mathbf{p}_{\text{-}1} + \mathbf{R}(\theta_\mathbf{s})\boldsymbol{\epsilon} \tag{3.6}$$

Where:

$$\mathbf{b}_n = \begin{bmatrix} b_x \\ b_y \end{bmatrix}_n \quad , \quad \mathbf{m}_n = \begin{bmatrix} m_x \\ m_y \end{bmatrix}_n \quad , \quad \mathbf{c}_n = \begin{bmatrix} c_x \\ c_y \end{bmatrix}_n ;$$
$$\mathbf{p}_{\text{-}1} = \begin{bmatrix} p_{\text{-}1_x} \\ p_{\text{-}1_y} \end{bmatrix} \quad , \quad \mathbf{p} = \begin{bmatrix} p_x \\ p_y \end{bmatrix} \quad , \quad \mathbf{p}_{+1} = \begin{bmatrix} p_{+1_x} \\ p_{+1_y} \end{bmatrix} .$$

In these representations, n corresponds to the number of sensor/data-sample and ranges from $1, \ldots, N_s$, where N_s is the total number of sensors/data-samples. Similarly, θ_s is the azimuth of the n^{th} sensor and $d(\theta_s)$ corresponds to the distance, which subsequently is scaled down by α_b to bring the bacterial activity to the vicinity of robot, in eq. 3.4. The subscripts -1 and $+1$ refer to previous and next instances, respectively. The $\mathbf{R}(\theta_\mathbf{s})$ is a rotation matrix, as under:

$$\mathbf{R}(\theta_\mathbf{s}) = \begin{bmatrix} \cos(\theta_s) & -\sin(\theta_s) \\ \sin(\theta_s) & \cos(\theta_s) \end{bmatrix}$$

At least one monocyte is associated with each sensor/data-sample. If a robot has a total of 16 photosensors, it will hold at least 16 monocytes, *virtually* within itself. Similarly, if a robot has a sonar, it can be sampled to construct a similar arrangement by using data-samples as sensors.

3.2.2 Phagocytosis

Once a monocyte finds chemoattractant molecules during its biased-random motion, it may move towards the bacterium and engulf it. This phenomenon is called phagocytosis [72]. In terms of its implementation, a stochastic approach is applied where a population of artificial monocytes is

associated with robot-morphology. The following discussion provides the details of two major components of phagocytosis: *chemotaxis* and *biased random walk*.

3.2.2.1 Chemotaxis

The steering agents within a robot undergo *chemotaxis*. It refers to the movement of organisms towards some chemicals in their immediate environment or away from some others. The bacteria, represented as **B** in the previous subsection, secrete chemicals that diffuse with time according to eq. 3.7. Similarly, the cytokines, released by monokines, also add to the combined value of chemoattractants and chemorepellents. The diffusion is implemented using neighborhood functions ($N_{(i,j)}$), with $w_{c_1} > w_{c_2}$.

$$C_{new}(i,j) = w_{c_1} C_{old}(i,j) + w_{c_2} C_{old}(N_{(i,j)}) \tag{3.7}$$

where, C refers to combined value of chemoattractant and chemorepellent molecules at lattice point (i,j) and its neighborhood $N_{(i,j)}$. The weights w_{c_1} and w_{c_2} define the diffusion rate. w_{c_2} is a vector of coefficients with elements equal to r in the neighborhood function. The neighborhood [6], on the other hand, is either a Von-Neumann ($N^v_{(i,j)}$ in eq. 3.8a), Moore ($N^m_{(i,j)}$ in eq. 3.8b) or a user-defined neighborhood.

$$N^v_{(i,j)} = \Big\{(x,y) : |x-i| + |y-j| \leq r\Big\} \tag{3.8a}$$

$$N^m_{(i,j)} = \Big\{(x,y) : |x-i| \leq r, |y-j| \leq r\Big\} \tag{3.8b}$$

3.2.2.2 Biased Random Walk

Once diffused chemicals are found around the periphery of the robot, steering agents (monocytes) move a biased random walk according to the gradient of combined chemoattractant-chemorepellent presence. Biased-random-walk [3] is based on probability (Pr) of going from current position ($\mathbf{p}$) to next position ($\mathbf{p}_{+1}$) according to the gradient (ΔC) of the chemoattractant molecules, with (N_s) number of sensors, as shown in eq. 3.9.

$$Pr(\mathbf{p} \rightarrow \mathbf{p}_{+1}) = \frac{1}{N_s} \pm \Delta C \tag{3.9}$$

For example, in a 2D biased random walk with chemoattractant gradient ($\Delta C_x, \Delta C_y$), this probability can be defined as under, using Von-Neumann neighborhood. A random number is generated for each monocyte. If the value comes out to be between 0 and $1/4 + \Delta C_x$, the monocyte moves to the +ve x direction and similarly for other values.

$$\begin{aligned} Pr(x \rightarrow x \pm 1) &= 1/4 \pm \Delta C_x \\ Pr(y \rightarrow y \pm 1) &= 1/4 \pm \Delta C_y \end{aligned} \tag{3.10}$$

Subsequently, a Monte Carlo methodology is applied on the monocyte-movement for each sensory reading. It refers to a class of computational methodologies that *repeat* their random samplings to find the results [69]. The question, however, is how to justify the use of repeated random samplings to move monocytes. It is because a robot should avoid accidents that may

result from a single random sample, despite a bias in the chemoattractant-gradient. The steering agents incrementally move a biased-random-walk in each iteration of the Monte Carlo simulation. The probability density of the resulting cluster of steering agents defines the future direction of the robot. It is, however, important to find the number of iterations a Monte Carlo simulation should execute. The number of required iterations is higher in case of purely-random or pseudo-random numbers. The monocytes, on the other hand, have a gradient-bias and can have a lower number of iterations. A test function was executed to find the number of required iterations for the Monte Carlo simulations. The function presents an obstacle-filled data to monocytes and records the collision-count and computational-time vis-à-vis the number of iterations. The selected number of iterations corresponds to a minimum of both the collision-count and the computational-cost. The results are shown fig. 3.8 & 3.9. The user also has the liberty to change the number if required.

The probability density of the monocytes in the robot's arena is computed using *kernel density estimation* method [81]. The method is usually employed on a finite data sample to infer results from a population. It is used in this research to find the maximum probability of monocyte-density in both x and y directions and consequent results are inferred as a steering direction of the robot. The results of monocyte-probability-density, on last instance of robot movement, are shown in fig. 3.3(b). The application of probability density estimation, on the results of aforementioned Monte Carlo simulations, results in finding the instantaneous target destination.

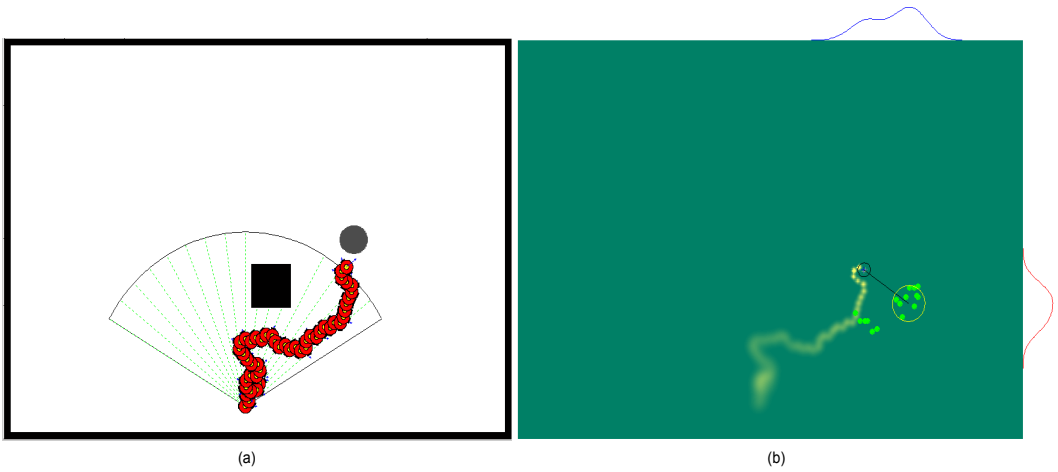


Figure 3.3: The robot, with $\left\{[-\pi/3, \pi/3],\ 16,\ \pi/24\right\}$ configuration, avoids the obstacle (black) and seeks the target (gray) using monokines as steering agents.

At this stage, the robot is able to navigate with a degree of randomness to ensure a probabilistic approach, albeit the robot-morphology. But what if the innate immunity proves to be insufficient to handle the assigned-task. A biological immune system uses *inflammation* as a feedback mechanism, in such situations. This research, therefore, implements an inflammation function, as under:

3.2.3 Inflammation

An inflammatory response is usually caused when first line of immuno-defense fails to hold a bacterial invasion. As a result, lymphocytic activity covers the bacterial surface with antibodies which, in turn, attract phagocytic cells to engulf the bacteria. This is one major explanation of an increased in-

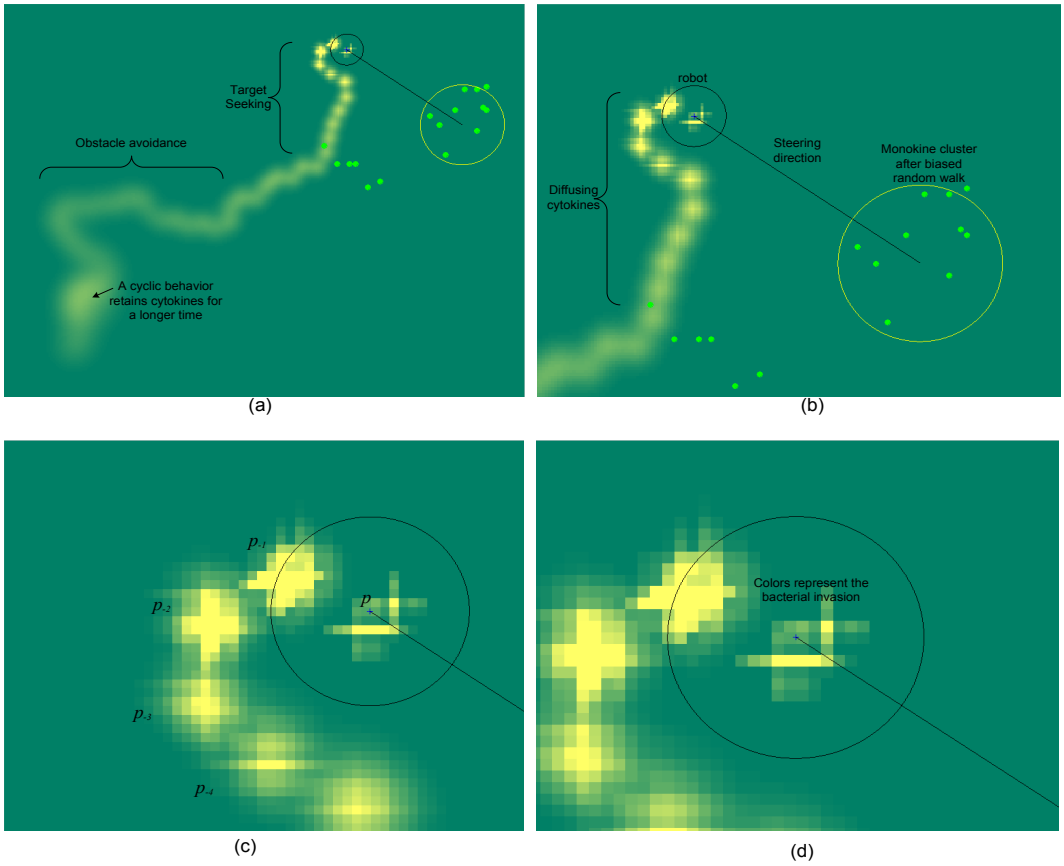


Figure 3.4: Details of innate immunity for robot navigation. (a) The trace showing obstacle-avoidance and target-seeking behaviors. (b) The steering direction towards the maximum probability density of monocytes. (c) Current and past positions in the trace. (d) Closeup of bacterial invasion in the robot.

flammatory response; other possibilities may also exist. This research uses the concept of inflammation to trigger different immuno-responses. Inflammation level increases when each robot within the AIS fails to successfully handle the situation e.g. *collision* increases and *rescue* decreases the inflammation in a simple SAR activity. Moreover, the maturity of dendritic cells is also linked

to the inflammation. Details of such a linkage is shown in eq. 3.16 and 3.17. The expression of inflammation (eq. 3.11) for a typical SAR scenario can be defined as under:

$$I = \left[w_1 \sum_{i=1}^{is} \kappa_i + w_2 \sum_{i=1}^{is} \tau_i - w_3 \left(\frac{\tau_r}{\tau} \right) \right] \tag{3.11}$$

Although, the inflammation function is application-specific, its abstraction is done on the basis of sustenance of harmful factors in recent past. In the expression above, inflammation will rapidly rise if collision (κ) is repeatedly experienced or a target (τ) is detected persistently. Additionally, inflammation level will reduce if targets are rescued (τ_r) in the arena. There can be other factors included in the expression with a provision that inflammation levels are normalized with in [0,1].

3.2.4 Dendritic cells

Monocytes can also develop into dendritic cells. These cells undergo a *maturation* on the basis of collected signals. The maturation-output suggests that other lymphocytes should also be mediated. This contextualizes the antigenic data which is consequently presented to the naïve T-cells, if dendritic cell maturity indicates *danger*.

Three types of external signals can be abstracted in view of the *danger theory* of immunology [53]. These are pathogen associated molecular patterns (**P**), danger associated molecular patterns (**D**) and safe signals (**S**). There can be a number of possible ways to define these signals. The following method-

ology is devised such that takes into account the robot-embodiment in this definition.

$$\mathbf{P} = [P_1, \ldots, P_{N_s}] \tag{3.12}$$

$$\mathbf{D} = [D_1, \ldots, D_{N_s}] \tag{3.13}$$

$$\mathbf{S} = [S_1, \ldots, S_{N_s}] \tag{3.14}$$

Where **P**, **D** and **S** are the vectors of dendritic-cell-signals corresponding to each sensor/data-sample. The length of these vectors is the same as the lengths of monocyte matrix (**M**).

It is noteworthy that these signals are either a result of mapped sensor-readings (e.g. P_n, D_{n_e}) or a consequence of internal inflammation function (e.g. D_{n_i}, S_n). Since P_n depends on presented pathogens, the corresponding expression (eq. 3.15) makes use of sensed distance information (d). Danger signals are defined in terms of both external and internal contexts, as shown in eq. 3.16. External danger (D_{n_e}) corresponds to frontal direction being more biased towards danger. Internal danger (D_{n_i}), on the other hand, is defined in terms of internal state monitored through inflammation (I). If the current inflammation level is greater than the previous reading, the danger is incremented in the danger-vector and vice versa. Similar is the case of each safe signal (S_n) which is recorded in the safety-vector (**S**), on the basis of lower inflammation than the preceding reading, as shown in eq. 3.17.

$$P_n = \frac{d(\theta_s)}{d_{\max}} \tag{3.15}$$

$$\begin{aligned} D_n &= [D_{n_e} \; ; \; D_{n_i}] \\ &= \left[\frac{1 - \cos(\theta_F - \theta_s)}{2} \; ; \; D_{n_i} = \begin{cases} 1, & \text{if } I_t > I_{t-1}; \\ D_{n_i}, & \text{if } I_t = I_{t-1}; \\ 0, & \text{if } I_t < I_{t-1}. \end{cases} \right] \end{aligned} \tag{3.16}$$

$$S_n = \begin{cases} 1, & \text{if } I_t > I_{t-1}; \\ S_n, & \text{if } I_t = I_{t-1}; \\ 0, & \text{if } I_t < I_{t-1}. \end{cases} \tag{3.17}$$

Greensmith's expression to compute the output from DC maturity in eq. 2.6 is modified according to the devised representation scheme for HMRS, as indicated in eq. 3.18. It uses a three signal specification as in Greensmith's DCA but differs in terms of fused data representation of each signal by specifying the lengths of signal vectors equal to the number of sensor readings around the periphery of the robot. Moreover, instead of using a three-step accumulation of signals, a window-size (ws) is specified to accumulate all the signals. The output ($\mathbf{O}$) consequently arbitrates the immunity levels.

$$\mathbf{O} = \beta \left[W_{\mathbf{P}} \sum_{i=1}^{ws} \mathbf{P} + W_{\mathbf{D}} \sum_{i=1}^{ws} \mathbf{D} + W_{\mathbf{S}} \sum_{i=1}^{ws} \mathbf{S} \right] \tag{3.18}$$

In an HMRS, dendritic cells do not translate into the steering directions as other monocytes do. They develop into mature-dendritic-cells if the output is above the migration threshold and consequently trigger the adaptive immunity level to counter the *dangerous* situation. Additionally, the polarity of chemoattractants, secreted during phagocytosis, is reversed in the window.

This reversal transforms the chemoattractants into chemorepellents and help other robot(s) to avoid similar situations. On the other hand, if the output is less than the migration threshold, robot remains within innate immunity realm. Moreover, the maturity of DCs depends on the accumulated signals as well as the window size.

3.2.5 Initiation of T-cell Maturity

The most important task of dendritic cells is to present contextualized antigen to the adaptive component of immunity. They are, therefore, known as antigen-presenting-cells (APC) as well. The cytokines secreted by matured dendritic cells as well as the co-stimulation (CSM) signal interact with Naïve T-cells, as illustrated in fig. 2.1. This research uses the concept of helper-T cells to avoid collisions and conserve/seek energy through negative and positive selections, respectively. The details are presented in the next chapter, where energy $E_T(\theta)$ and collisions $C_T(\theta)$ are computed on the basis of eq. 4.6 and 4.7, respectively.

3.3 Implementation of Innate Immunity Module

The preceding details of mathematical abstractions are only for the innate immunity module. The framework also has an adaptive immunity module to help robot(s) in their tasks. Once all the computations are performed either through innate or adaptive components, the robot(s) are steered using the expression in equation (4.3). At this level, only the first option is invoked

as a steering direction with maximum probability (Pr_1), through its steering agents ($\mathbf{M}$). Other options are explained in the next chapter. For now, before using (4.3), we select the steering using:

$$\theta_T = \begin{cases} \theta_i = \arg\max_{\theta} Pr_1 \ , & \text{if innate immunity;} \\ \theta_t = \arg\max_{\theta} Pr_2 \ , & \text{if adaptive immunity, T-cell level} \\ \theta_a = \arg\max_{\theta} A_i^r \ , & \text{if adaptive immunity, B-cell level} \end{cases} \tag{3.19}$$

Figure 3.5 shows multiple simulation runs with the same experimental configuration. It indicates that each run produces a different trajectory of robot's movement. This is the result of the non-deterministic nature of innate immunity module which enhances the capability of the robotic system. If a robot needs to *search* a target in a SAR system, it should explore the arena non-deterministically. A deterministic approach, on the other hand, may not find a target in the robot's *blind-spot* because it follows the same path, over and over again. Although, dynamic environments induce an effect of non-determinism in robot's movement, even if the underlying navigation scheme is deterministic. A probabilistic navigation approach by-passes the deterministic or pseudo-non-deterministic methodologies by presenting a *possibility* of different behavior in re-occurring conditions.

Similarly, fig. 3.6 illustrates robot-trajectories with three different morphological configurations. The span of localizing sensors is different in each case along with the number of steering agents. The resulting trajectories indicate that the innate-immunity module is able to avoid an obstacle and seek a target. Although, simulations indicate the successful nature of underlying

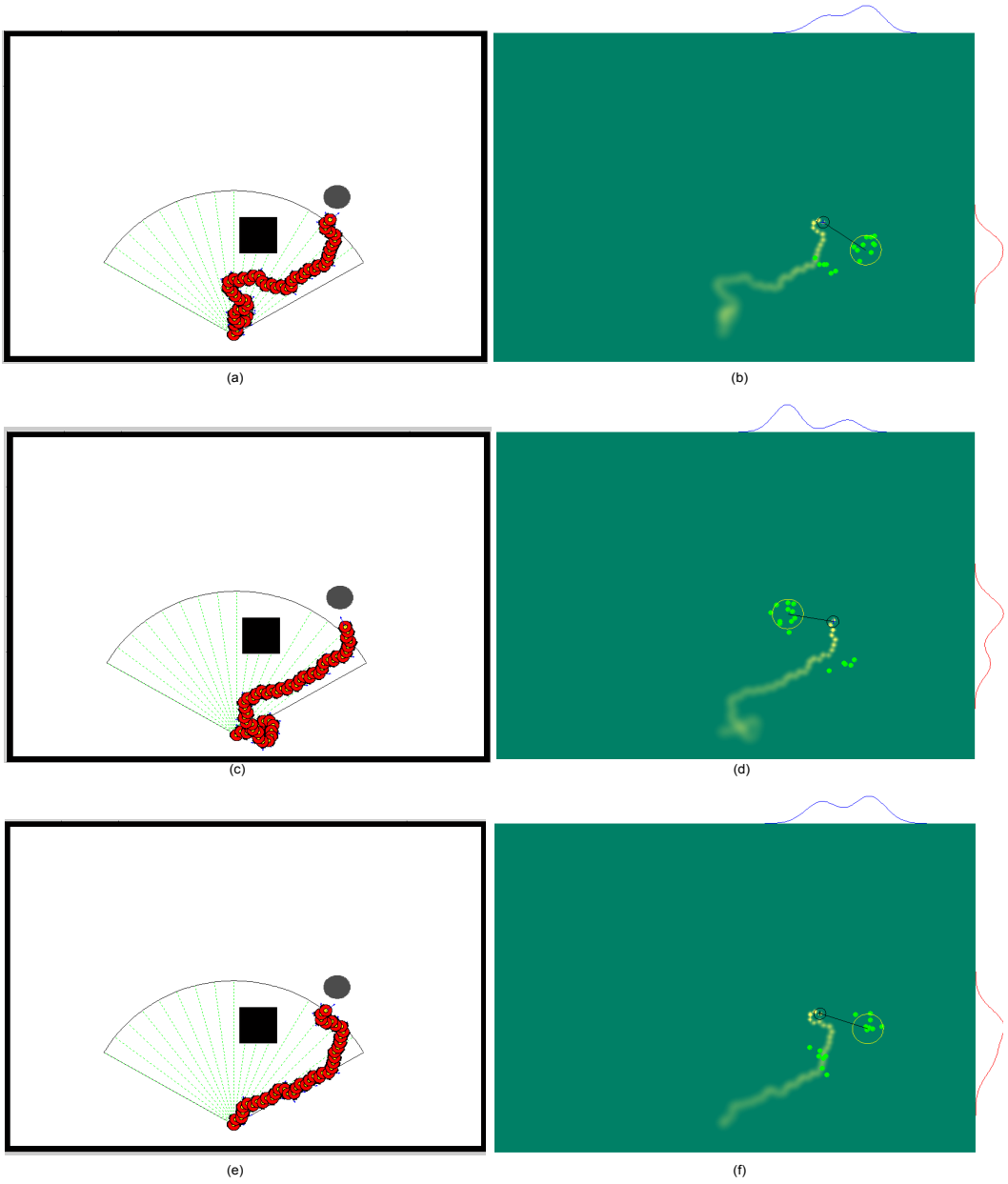


Figure 3.5: Multiple simulation runs of innate component of immunity with same robot-configuration and number of monokines

algorithm, the wobbling of robot's trajectory may not be required in some cases e.g. in case of an already-spotted-target, the *wobble* may not be ideal. The robot should move directly towards the target. The adaptive component, in such cases, offer the desired results. It is, therefore, important to combine innate and adaptive immunities together.

3.3.1 Algorithm

The algorithm of the framework's innate component is presented in table 3.2. The underlying mathematical details have been described in previous sections. It is, however, important to highlight that the system is designed to be both *distributed* and *cooperative*. The innate component is mainly distributed where each robot in the arena can move independent of other robots, as far as some navigation trace is not rendered as *repellent*. The adaptive component, on the other hand, ensures a cooperation, as shown in its algorithm in table 4.2.

3.3.2 Iteration count of Monte Carlo simulation

The test-function to decide the iteration count of Monte Carlo simulations is designed to minimize the expected collisions and computational cost. The algorithm of innate immunity is tested in two situations: one is the symmetric scenario where an obstacle is directly in front of robot's *face* and second is the skewed scenario in which the robot is not directly facing the obstacle. The details are shown in fig. 3.7. A collision is *expected* if robot's steering di-

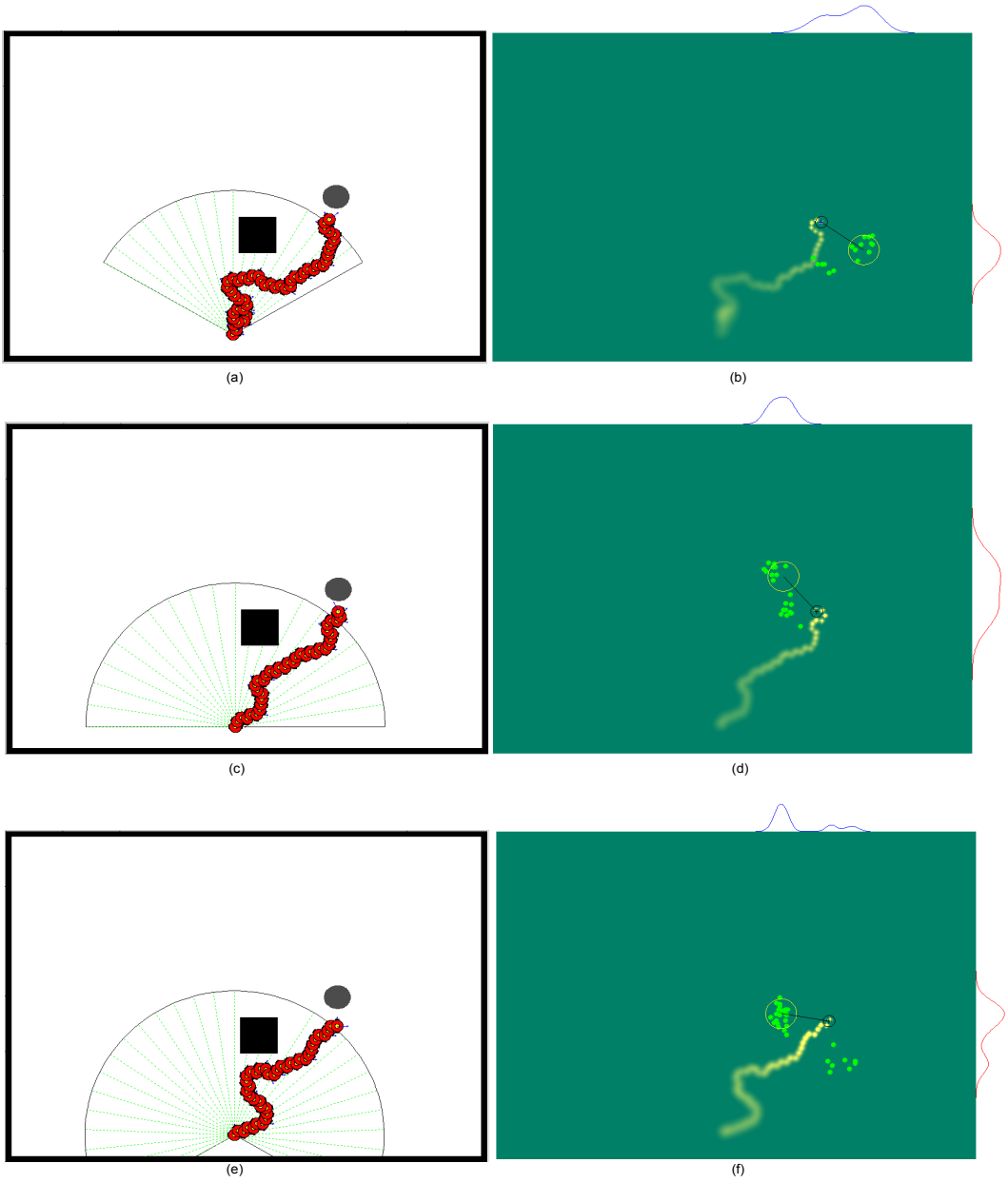


Figure 3.6: Multiple simulation runs of innate component of immunity with different robot-configurations and number of monokines

Table 3.2: Algorithm for innate component of immunity for HMRS

Algorithm

Input: Sensor data and external signals
Output: Actions and contextualized antigen
Initialize: Monocytes, Naïve T-cells

- **For** each robot **do**
  - **if** CSM output signal $<$ migration threshold **do**
    - Collect the instantaneous sensory data
    - Map sensor data as $\mathbf{B}$, $\mathbf{P}$, $\mathbf{D}_e$ & antigen
    - Diffuse the chemoattractants/chemorepellents
    - Compute the gradient
    - Walk the monocytes a biased random walk
    - Perform Monte Carlo simulations
    - Estimate/compute probability density of monocyte-locations
    - Move towards $\theta_i = \arg\max_{\theta} Pr_1$
    - Update inflammation (I)
    - Update P_n, D_n & S_n signals (internal & external)
    - Calculate the cumulative dendritic-cell output ($\mathbf{O}$)
    - Update cell location to lymph-node
    - **if** semi-mature output $>$ mature output **then**
      - environment is contextualized as safe
    - **else**
      - environment is contextualized as un-safe
    - **end**
    - Perform metadynamics
  - **else-if** CSM output signal $\geq$ migration threshold **do**
    - Invoke adaptive immune response
  - **end**
- **end**

rection, after its computations, is towards the obstacle. Such navigation does not necessarily indicate a collision but a possibility of it, if robot *continues* to travel on the same trajectory. The testing was repeated 20 times with same

experiment configuration and the results are presented in fig. 3.8 & 3.9. The number of iterations for an HMRS can be altered by a user. The simulations, however, use N_{it} to be 20, unless mentioned otherwise.

Figure 3.7 also shows the effects of changing the neighborhood r. The resulting gradient of chemoattractants/chemorepellents can be different. In simulations shown in this chapter, r is selected to be 3. The rationale of selecting its value is based on step-size of monocyte movement. If the step-size of monocyte movement is 1, r can be 1 or greater. If one selects the step-size to be greater than r, there remains a possibility of un-detecting an instantaneous sensory information.

This concludes the discussion on innate component of the framework. The aforementioned results are included only to illustrate the basics of innate immunity module; an exhaustive experimentation with complex scenarios is presented in subsequent chapters.

3.4 Chapter Summary

In this chapter, a detailed account of mathematical abstraction of innate immunity is provided for a mobile robotic system. The sensory data is mapped as bacteria. The steering agents of each robot, virtually housed within, are modeled as monocytes that move in a biased-random manner, towards the bacteria. The probability density of these monocytes, after Monte Carlo simulations, defines the next steering direction. The sensory data is represented in a manner that supports the morphological heterogeneity of different robots and

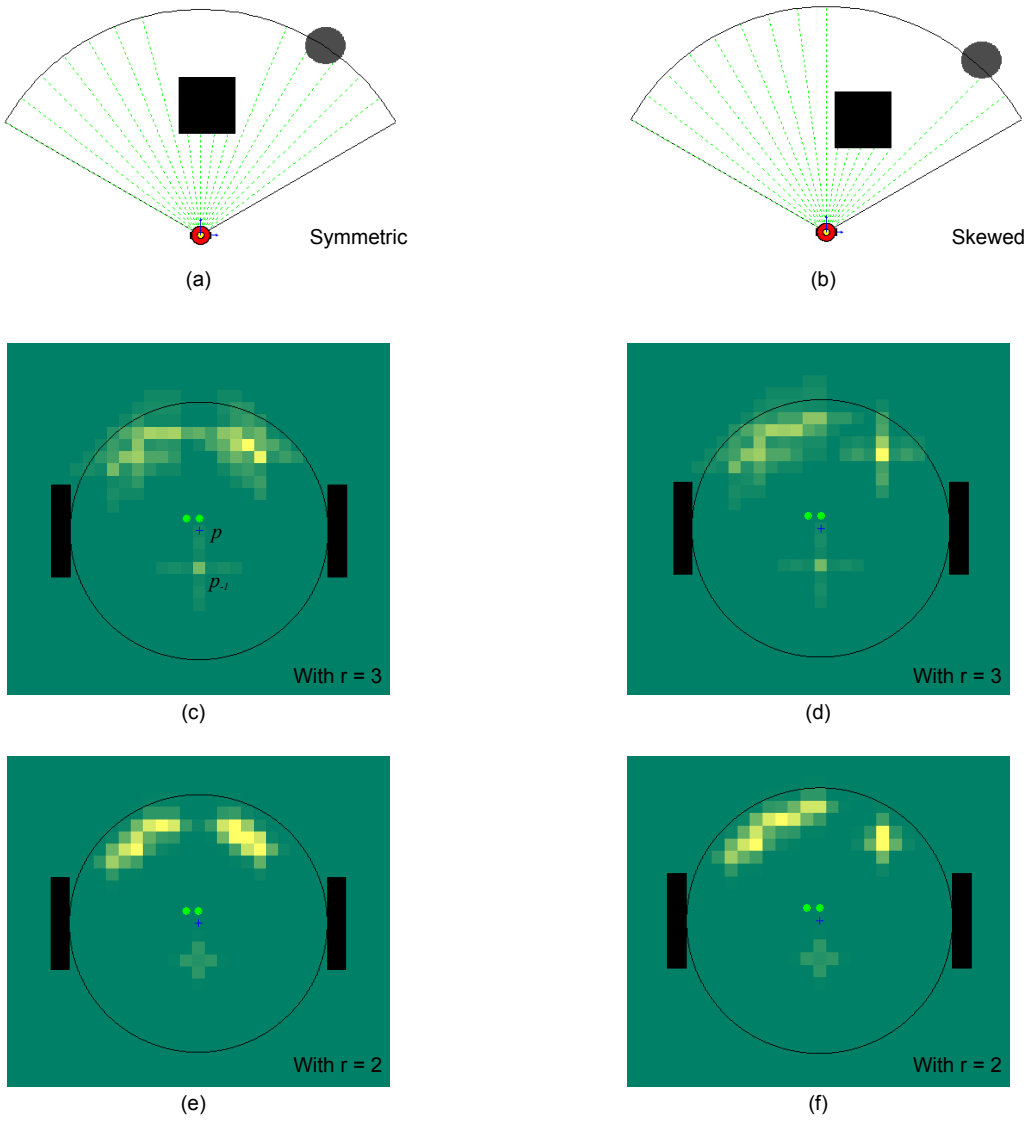


Figure 3.7: Symmetric and skewed scenarios with different neighborhoods

subsequent structure of T & B lymphocytes. Moreover, the previous traces of a robot's movement diffuse and consequently help in avoiding future occurrences

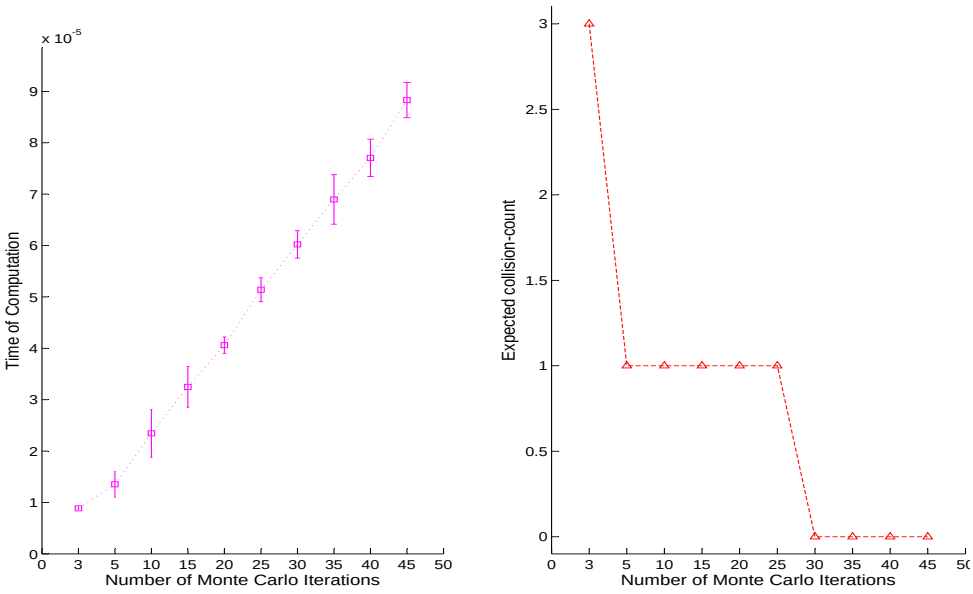


Figure 3.8: Results of test-function to calculate the number of required Monte Carlo iterations for a skewed scenario

of cyclic/bad behaviors. The diffused chemoattractants/chemorepellents are maintained within the system to help other robots as well, for a period specified by the diffusion rate. An internal feedback mechanism of inflammation is also incorporated that helps in dendritic-cell-maturity and consequent invocation of adaptive immunity.

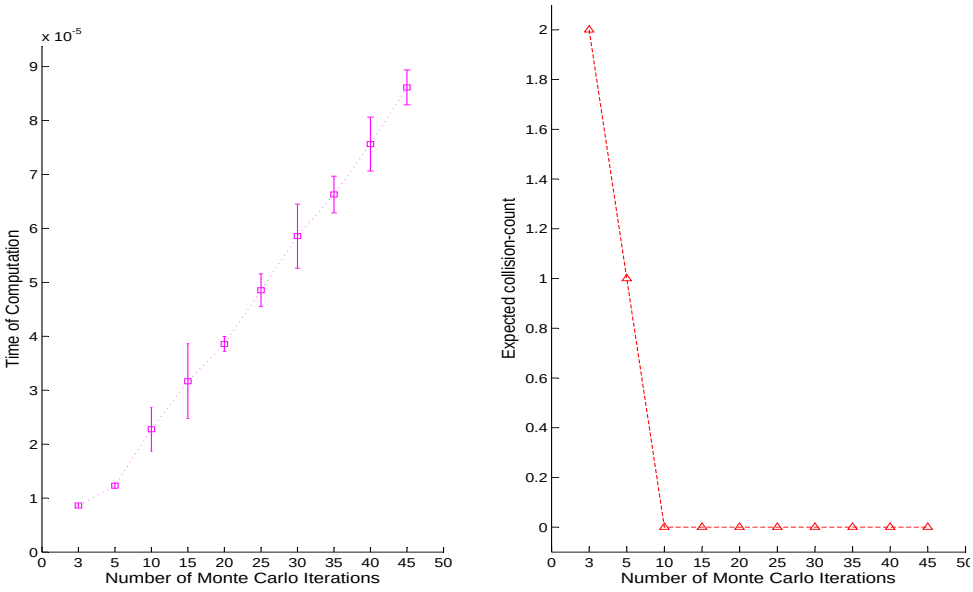


Figure 3.9: Results of test-function to calculate the number of required Monte Carlo iterations for a symmetric scenario

Chapter 4

Adaptive Component of the Framework

4.1 Inspiration from Adaptive Immunity

Once innate immunity fails to maintain homeostasis, it calls for help through co-stimulation. The adaptive immunity responds with its capability to *evolve* through cloning and hyper-mutation. This consequently *unlocks* the antigenic code to handle the bacterial/viral *attack*. Similarly, once a robot fails to handle a situation/task, it calls for *help*. The *artificially* intelligent system, if present, responds with a capability to evolve suitable actions. This consequently handles the situation. A robotic system can, therefore, be designed through biological inspiration. One can appreciate the metaphorical similarity but the adaptive component of immunity offers more; it performs positive/negative selection through T-cells, stimulates/supresses the antibodies by maintaining an idiotypic network and maintains a repertoire of successful cells in its memory. Although, there can be different ways to translate the immunological working to a robotic system, this research aims to develop a framework by combining the innate component, designed in the previous chapter, with its adaptive counterpart. The metaphors are detailed in the following table 4.1. The metaphors work both within a robot (complex "brain") or in a fleet. Another level of metaphors emerges when multiple robots are involved

in executing the task in a cooperative fashion.

Table 4.1: Metaphors for Adaptive Component of Immunity-based HMRS.

Adaptive Immune System	**Robotic System**
Antigen	Environment around a robot through instantaneous sensory information
Antibody	Actuatory responses of the system (steering & specialty actuations)
T lymphocyte	A mechanism to generate actuatory response
Cytotoxic T cells	Steering agents for robots' movement
Helper T cells	Triggering agents for evolutionary mechanism
B lymphocyte	An evolutionary mechanism waiting to be triggered
Plasma B-cell	A mechanism to evolve actuatory responses
Idiotypic Network	A mechanism to stimulate successful actions
Clonal Selection	A mechanism to select successful actions after cloning & hypermutation
Affinity	Distance between mapped environment and possible actions

This chapter is aimed to augment the innate component of AIS by maturing T & B lymphocytes as an adaptive critic and effector, respectively. B-cell maturity results in plasma and memory cells. Plasma cells, resultantly, evolve and regulate the antibodies according to the CS-theory and are translated into robotic actions. Immune-memory, on the other hand, is constituted with matured B and T lymphocyte-populations and serve as a repertoire of successful actions. The Idiotypic Network (IN) is maintained by translating robot-heterogeneity into binary and navigation-directions into real component of antibody-structure. The network also ensures a communication mechanism in which antibodies with higher concentrations are communicated back to the

respective site of pathogen invasion.

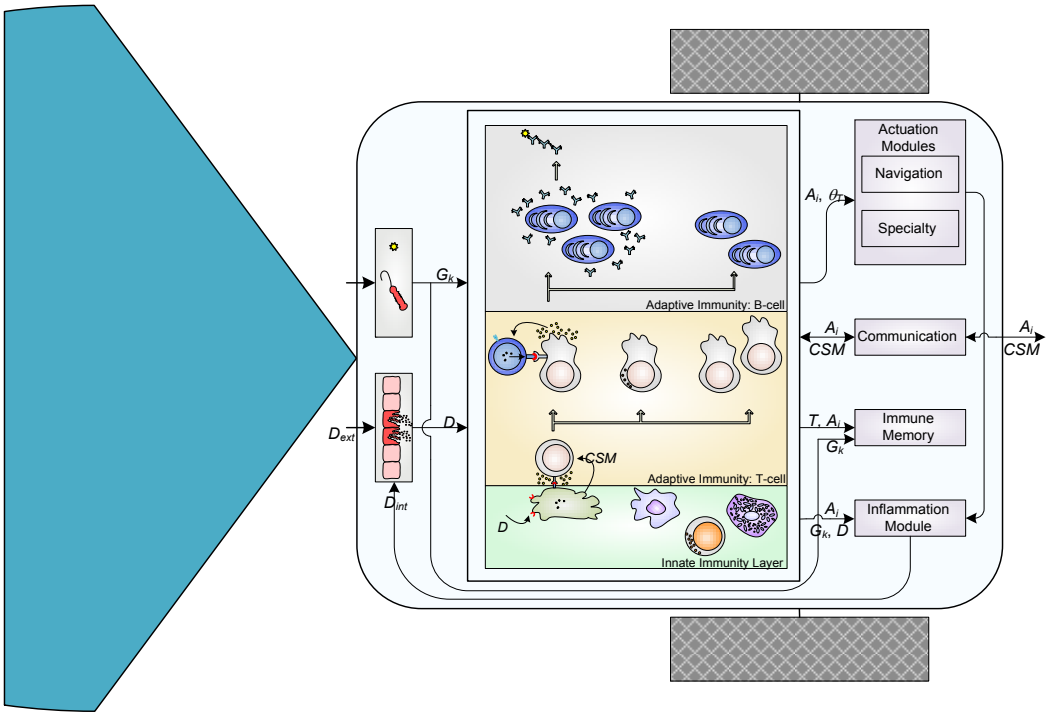


Figure 4.1: A generalized structure of robot's AIS in an HMRS, combining innate and adaptive components

Figure 4.1 illustrates a robot with innate as well as adaptive components of immunity. The robot can communicate the co-stimulation molecule (CSM) to its adaptive immunity layer, if innate immunity fails to maintain a low inflammation level. Similarly, other robots in the arena/theater can also communicate with the robot through CSM. The adaptive layer, consequently, undergoes its functions to output the antibodies to its own and/or others' actuation modules.

4.2 Mathematical Abstraction

In order to translate the adaptive immunity in an HMRS, while avoiding a temptation to *completely* replicate the biological details, Castro's guidelines [15] offer a starting point, as shown earlier in fig. 2.3. The following subsections, therefore, provide the mathematical details of the framework's adaptive component including representation, T & B lymphocyte activity, idiotypic networking and maintenance of memory.

4.2.1 Representation

The representation scheme uses a hybrid approach, a combination of binary and real strings, to translate sensory data into antigens and network data into antibodies. It is designed to act independently if a user decides to use the adaptive component only and to integrate with innate immunity if one decides to experiment with a combined approach.

As a first step, the environment is translated into antigen (G_k). The problem, however, is to deal with sensor-heterogeneity. Sensors can be subdivided into two categories according to their functionality: one to sense the local environment (including self – i.e., internal variables, like battery state-of-charge, SoC, current drawn by motors, etc.) and second to collect speciality data. The local environment is sensed through photosensors, proximity sensors, cameras, sonars, etc. and can be represented generically. Speciality sensors, on the other hand, can be of varied nature like color sensors, smoke detectors, accelerometers, seismometers, to name a few, and cannot be con-

fined in a generic representation. These can, however, be merged into existing localizing data (like SLAM – Simultaneous Localization And Mapping), if azimuthal orientation of both types of sensors can be matched. Alternatively, data from speciality sensors can be treated as a separate antigen into the network. The detail of such situations is illustrated in fig. 4.2 in which localizing and speciality data is mapped as equal length bit-strings which can consequently be merged as a single antigen. The bit-string of epitope can be visualized as a bi-directional data representation, centered at robot's frontal direction (θ_F).

In the case of photosensors, for example, their location on the robot periphery (θ_s) is represented along with the sensed data (S). In the case of sonar or camera, however, sensed data is sampled along the range of these sensors, e.g. if a sonar senses 360° around a robot, this data can be sampled into 360 instances, each 1° apart. In this way, a single antigenic representation can be used, as indicated in eq. 4.1, albeit the respective robot-morphologies. A detailed account of antigen-mapping is illustrated in fig. 5.5. It is to note that the length of each bit-string is different for each robot-morphology and is indicated in eq. 4.1 as n. It signifies that a flexible string representation scheme is devised, in contrast to the fixed length data strings in reported immunity-based robotic applications.

The second step is to represent antibodies (A_i) using a hybrid approach. The devised structure of antibodies has three components. The first component of each antibody is a fixed *binary* string which refers to (a) particular

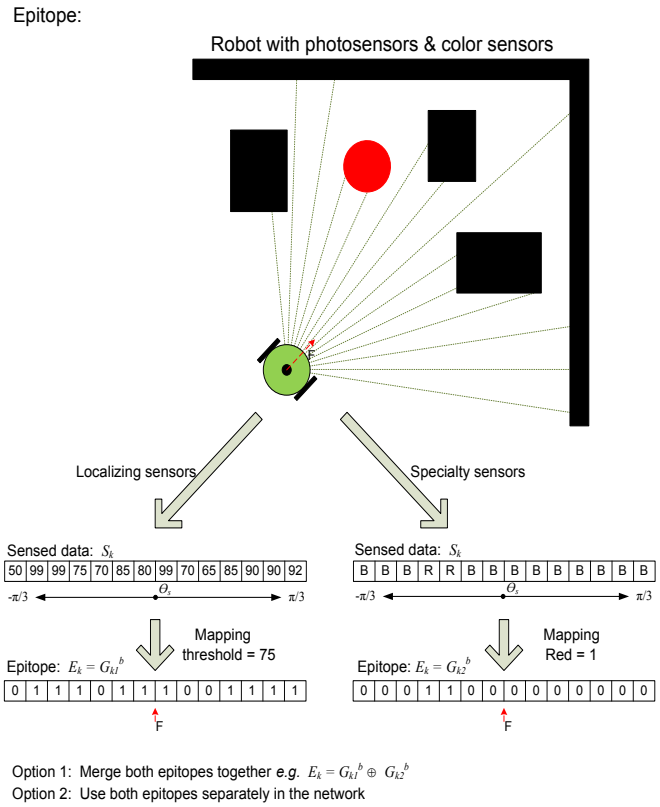


Figure 4.2: Structure of epitopes in antigen representation. The illustrated robot, with $\left\{[-\pi/3, \pi/3],\ 14,\ \pi/21\right\}$ configuration, is fitted with photosensors as localizing and color-sensors as speciality sensors.

robot(s) in the system (R_i). The second component is a flexible *binary* string for paratope/idiotope representation. The third component is *real* which contains the morphological information of the robot and corresponds to the steering direction of respective robot(s). The structural detail of paratope (P_i) or idiotope (I_i) is defined in eq. 4.2 and illustrated in fig. 4.3. Moreover, the stimulation and suppression of antibodies resulting from paratopes and idiotopes

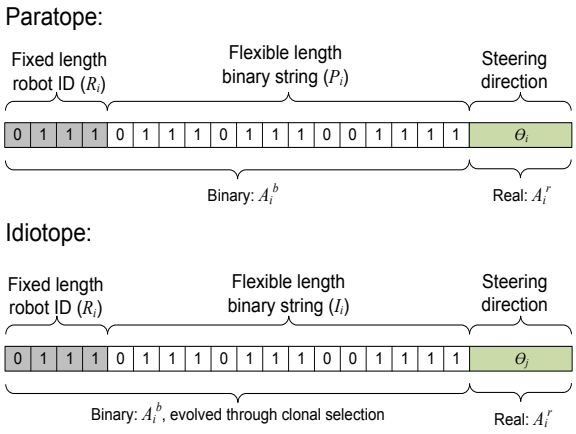


Figure 4.3: Structure of paratopes and idiotopes in hybrid idiotypic network. In this illustration, number of bits in fixed length robot-ID ($N_{bits_{R_i}}$) is 4, number of bits in flexible length binary string (N_a) is 14 and $i, j = 1, \ldots, N_a$.

is illustrated in fig. 4.4.

$$\begin{aligned} G_k &= \left\{G_k^b \ , \ G_k^r\right\} \\ &= \left\{\{0,1\}_k^n \ , \ \{\theta_s \ , \ S\}_k\right\} \\ &= \left\{E_k \ , \ \{\theta_s \ , \ S\}_k\right\} \end{aligned} \tag{4.1}$$

$$\begin{aligned} A_i &= \left\{A_i^b \ , \ A_i^r\right\} \\ &= \left\{\left\{R_i \ , \ \{0,1\}_i^{N_a}\right\} \ , \ \pm\frac{\pi}{N_a}(i-1)\right\} \\ &= \left\{\left\{R_i \ , \ P_i\right\} \ , \ \pm\frac{\pi}{N_a}(i-1)\right\} \end{aligned} \tag{4.2}$$

Superscripts b & r refer to binary and real parts of antigen/antibody, respectively. Once the network recomputes the concentrations of antibodies, the antibody with maximum concentration is selected as the output. The

binary component of the output, consequently, selects a robot- or a speciality-behavior, whereas the real component gives the steering directions to each robot, according to eq. 4.3.

$$\theta_T = \arg\max_{\theta} A_i^r \ , \quad \text{for robot selected by } A_i^b \tag{4.3}$$

4.2.2 T Lymphocytes

T lymphocytes are important because they upgrade the immuno-responses through co-stimulation on one end, and on the other, help the B-cells to mature an multiply. Naïve T-cells mature and multiply in the thymus, then undergo positive/negative selection that results in a few escaping to regulate the adaptive immunity [93]. T-cells can mature into different types including helper, cytotoxic, memory and regulatory T-cells.

As a first step, the T-lymphocytes are defined in a manner similar to the declaration of monocytes, i.e., one T-cell for each sensor/data-sample

$$\mathbf{T} = [T_1, \ldots, T_{N_s}] \tag{4.4}$$

where

$$T_n = \Big\{ E_T - C_T \Big\}_n \tag{4.5}$$

Each cell in $\mathbf{T}$ (4.4) is defined as a result of positive and negative selections in terms of energy and collision, respectively. Energy (E_T) and collisions (C_T) are computed in each sampled direction as a weighted sum of sensory input,

defined as:

$$E_{T_n} = \left\{ E_f \cdot \varphi_n + E_r \cdot \gamma_n - E_m \cdot \mu_n - E_\omega \cdot \omega_n \right\} \tag{4.6}$$

$$C_{T_n} = \left\{ C_c \cdot \kappa_n \right\} \tag{4.7}$$

Here, E_f, E_r, E_m, E_ω, and C_c are rewards and penalties, whereas φ, γ, μ, ω and κ correspond to food, good move, normal move, wait and collision, respectively, as shown in eq. 4.8.

$$\begin{aligned}
\varphi_n &= \begin{cases} 1 & \text{if} \quad f_{\mathbf{p}_{+1}} \neq 0 \\ 0 & \text{otherwise} \end{cases} \\
\gamma_n &= \begin{cases} 1 & \text{if} \quad \mathbf{p}_{+1} \neq \mathbf{p} \\ 0 & \text{otherwise} \end{cases} \\
\mu_n &= \begin{cases} 1 & \text{if} \quad \neg\texttt{obstacle} \\ 0 & \text{otherwise} \end{cases} \\
\omega_n &= \begin{cases} 1 & \text{if} \quad \mathbf{p}_{+1} = \mathbf{p} \\ 0 & \text{otherwise} \end{cases} \\
\kappa_n &= \begin{cases} 1 & \text{if} \quad \texttt{collision} \\ 0 & \text{otherwise} \end{cases}
\end{aligned} \tag{4.8}$$

The values of rewards and penalties can have other specifications according to the nature of robotic application. The specifications in eq. 4.8 correspond to a basic navigation problem in which a robot tries to wander, avoiding obstacles and searching food.

Reported literature on immunity-based robotic applications does not use the concept of T-cells in terms of energy conservation, collision avoidance or a similar behavior. The mechanism devised in this research uses T lymphocytes as steering agents, similar to the monocytic representation. The difference between monocytes and T-cells, as steering agents, is the definition

of the environment. In the case of monocytic representation, the sensed data is mapped in terms of bacteria which attracts the monocytes. The T-cell definition, on the other hand, uses the concept of *expecting* a gain of virtual-energy and an avoidance of virtual-collision.

The T lymphocytes also walk a biased random walk and undergo a Monte Carlo simulation. The resulting probability density is computed using kernel density estimation to find the future direction of robot, according to the following expression.

$$\theta_T = \arg\max_{\theta} Pr_2 \ , \quad \text{for robots in T-cell layer} \tag{4.9}$$

4.2.3 B Lymphocytes

B lymphocytes, once triggered by a *help* signal from helper-T cells, are defined as an evolutionary mechanism for HMRS to evolve suitable actions. Following the CS theory, the affinity between mapped environment and available actions maturates the artificial B-cells into plasma cells. Plasma cells, in-turn, secrete mutants of affinitive actions. Immune-memory retains some cells according to their affinities. The following subsections present the details of two major components of B-cell activity: clonal selection and idiotypic network.

4.2.3.1 Clonal Selection

The CS-component of the algorithm starts with the specification of initial solution candidate by randomly generating antibodies. Best antibod-

ies are selected on the basis of affinity evaluation. Affinity function uses the information of antibody & antigen to calculate the affinity. The results are consequently sorted in ascending order of their affinity. The antibodies are subsequently reselected on basis of best population size. It is followed by maturation and cloning of reselected antibodies on the basis of eq. 2.2 & 2.3, respectively. The clones are then projected within the solution bounds. The population size is estimated and subsequently used in reselection and reordering of clones. Selected-best-clones then replace the antibodies in initial antibody matrix.

4.2.3.2 Idiotypic Network

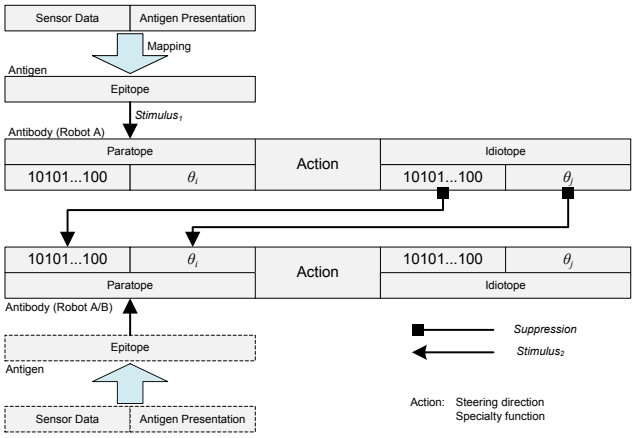


Figure 4.4: Structure of Antigen-Antibody and Antibody-Antibody interactions in an idiotypic network.

Idiotypic network resulting from aforementioned hybrid representation

is shown in eq. 4.10, by modifying the Farmer's equation (eq. 2.4).

$$\begin{aligned}\dot{A}_i &= \left[\alpha_a \left\{\sum_{j=1}^{N_a} m_{ij}^b a_j^b + \sum_{j=1}^{N_a} m_{ij}^r a_j^r\right\} - \alpha_s \left\{\sum_{j=1}^{N_a} m_{ji}^b a_j^b + \sum_{j=1}^{N_a} m_{ji}^r a_j^r\right\} + \sum_{k=1}^{N_g} n_{ik} g_k - \lambda_i\right] a_i \\ &= \left[\alpha_a \sum_{j=1}^{N_a} m_{ij}^b a_j^b - \alpha_s \sum_{j=1}^{N_a} m_{ji}^b a_j^b + \sum_{j=1}^{N_a} \cos(\Delta\theta_{ij}) a_j^r + \sum_{k=1}^{N_g} n_{ik} g_k - \lambda_i\right] a_i \end{aligned} \tag{4.10}$$

where,

$$m_{ij}^b = A_i^b \rightarrow A_j^b = \sum_{l=1}^{L} (I_i(l) \oplus \overline{P_j(l)}) \tag{4.11a}$$

$$m_{ji}^b = A_i^b \leftarrow A_j^b = \sum_{l=1}^{L} (I_j(l) \oplus \overline{P_i(l)}) \tag{4.11b}$$

$$n_{ik} = G_k^b \rightarrow A_i^b = \sum_{l=1}^{L} (E_k(l) \oplus \overline{P_i(l)}) \tag{4.11c}$$

The binary components of eq. 4.10 (m_{ij}^b, m_{ji}^b & n_{ik}) are designed after Hamming shape space [21, 24], where P_i, I_i & E_k correspond to paratopes, idiotopes and epitopes, respectively. These are evolved using clonal selection that, in turn, results in selection of a suitable robot. The real component of the idiotypic network (IN), on the other hand, is designed using combined stimulative-suppressive effect of Luh's expression [50]. The IN structure is also detailed in fig. 4.4. The whole procedure of using immunity, in a nutshell, makes a search-robot call for a rescue-robot in a situation where it is unable to pick up a subject, resulting in changing the paratopes' binary structure, and consequently evolving the idiotopes to increase antibody-concentrations for rescue robot(s). Furthermore, the real part of *instigated* antibodies corre-

spond to navigation, thus making rescue robot(s) move towards the subject to be rescued.

The network updates the antibody-concentrations in each instance of robot motion. Figure 4.6 indicates the antibody concentrations with zero initial conditions, in conjunction with robot position in fig. 4.5. It is evident that the stimulations and suppressions of idiotypic network result in increasing the concentrations of some antibodies. The robot, consequently, moves in the direction of these antibodies, using an elitist selection approach.

4.2.4 Immune-memory

The memory structure of the framework contains two subsets: one lists a population of T-cells with corresponding antigens and the other maintains a number of matured B-cells along with a successful G_k-A_i structure. At the start of a simulation, the memory matrix is empty but as the simulation proceeds, successful lymphocytes fill-up the empty spaces. Moreover, this memory is not static because the populations, maintained within, are continuously updated according to the system's meta-dynamics. It means that if a robot has moved out of a mine-field, the memory would gradually remove the cells (apoptosis) responsible for mine detection, up to a "small" population level. If in future, the robot stumbles again into a mine-field, the small population of corresponding cells would start increasing the relevant antibody populations.

4.3 Implementation of Adaptive Immunity Module

4.3.1 Algorithm

The algorithm of framework's adaptive component is presented in following table 4.2. The underlying mathematical details have been described in previous sections. It is, however, important to highlight that the system resorts to the adaptive component when innate component (table 3.2) fails to maintain homeostasis. It has two major components nested into each other: the first is the T-lymphocyte component whereas the second component corresponds to B-lymphocytes.

4.4 Chapter Summary

A detailed account of the mathematical abstraction of adaptive immunity is provided in this chapter. The sensory data, mapped as *antigen*, is already contextualized because of the *innate immunity module*. Once invoked, the *steering agents* of each robot are modeled as *T-lymphocytes* that move in a biased-random manner, according to positive and negative selections of *energy* and *collisions*, respectively. The probability density of T lymphocytes, after Monte Carlo simulations, defines the next steering direction. If inflammation continues to rise, the *adaptive immunity* jumps from *T-cell* to *B-cell* level. The B-cell layer responds by evolving *actions* through cloning and hyper-mutation. This is followed by network dynamics to output the concentrations of actions (*antibodies*). The antibodies with highest concentrations are selected as actuation signals to each robot in the system. Moreover, the previous traces of a

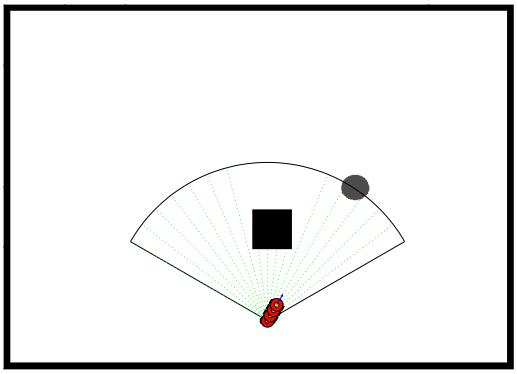
Table 4.2: Algorithm for adaptive component of immunity for HMRS

Algorithm

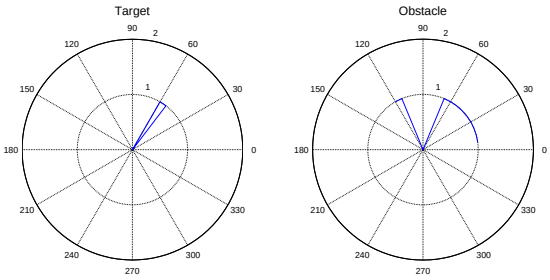
Input: Antigen, Antibody definition
Output: Evolved antibody network A_i, Memory: matured T & B-cells

- **While** CSM output signal $\geq$ migration threshold **do**
  - maturate & multiply naïve T-cells
  - Conduct positive selection
  - Perform negative selection
  - Secrete & diffuse cytokines from resulting helper-T-cells
  - Walk T lymphocytes a biased walk
  - Perform Monte Carlo simulations
  - Estimate/compute probability density of T-cell locations
  - **if** Help signal $\geq$ migration threshold **then**
    - **for** all Antigens G_k **do**
      - Compute affinities (stimulations & suppressions)
      - Select A_i & order
      - Clone and mutate A_i
      - Reselect clones
      - Perform network dynamics
      - Perform metadynamics
    - **end**
    - Move towards $\theta_a = \arg\max_{\theta} A_i$
  - **else**
    - Move towards $\theta_t = \arg\max_{\theta} Pr_2$
  - **end**
  - Update inflammation, CSM, Help & memory
  - **repeat**
- **end**

robot's movement continue to diffuse and consequently help in avoiding future occurrences of bad behaviors.



(a)



(b)

Figure 4.5: (a) The robot, with $\left\{[-\pi/3, \pi/3],\ 16,\ \pi/24\right\}$ configuration, on its way to avoid the obstacle (black) and seek the target (gray). (b) The sensor reading at last instance

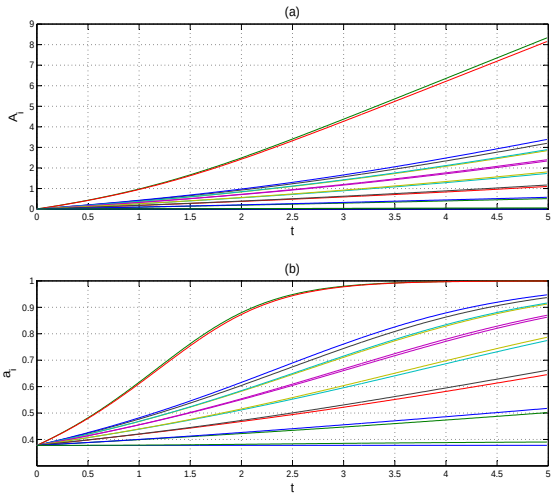


Figure 4.6: Antibody concentrations of the robot, with $\left\{[-\pi/3, \pi/3],\ 16,\ \pi/24\right\}$ configuration, during one instance of its motion

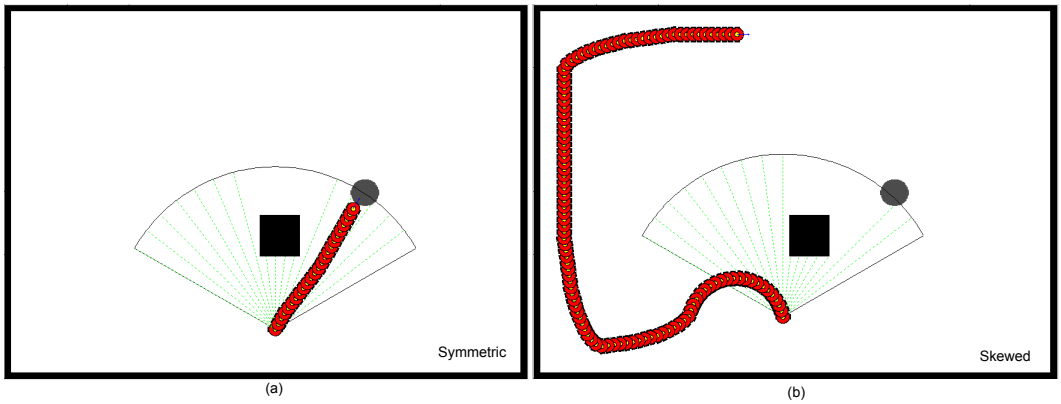


Figure 4.7: (a) Symmetric starting scenario, (b) Skewed starting scenario, illustrating the possibility of skipping the target

Chapter 5

Experimentation

Extensive experimentation is presented to establish the validity of Immune-Based Framework (IBF) for HMRS (Heterogeneous Mobile robotic System). In a typical multi-robot SAR (Search-And-Rescue) scenario, an experiment is termed as *successful* if the search-robot(s) successfully search(es) *all* the randomly placed targets, avoiding obstacles/traps, and subsequently calls the appropriate rescue-robot(s) to complete the mission. The design of an appropriate experiment, thus, requires to build an environment that exhibits the underlying capabilities of a robotic system. The requirement of an *unstructured* environment also demands that targets and obstacles be placed randomly and their knowledge is not communicated to the robots, *a-priori*. Moreover, heterogeneity is required to be translated in terms of a set of robot(s) configurations according to the morphological details in fig. 2.6. In this research, therefore, an experiment is configured on three levels: first by configuring the arena, second by configuring the robot(s) with different sensors and actuators and third by selecting the performance metrics to record the relevant experiment data. The details are presented below.

5.1 Scenarios

Although the major task is to "search-and-rescue" using multiple robots, single robot navigation is also included in the experimentation in order to establish a comparison with other AIS-based techniques. It is observed that a limited experimentation using a few pre-selected scenarios is insufficient to establish the validity of a robotic system e.g. a foraging robot of Ishiguro-Watanabe stream in a simple environment is insufficient to establish the cogency of idiotypic network approach. A more recent finding of the European research project BRICS [61], on best practices in robotics, suggests a rigorous experimentation for any robotic algorithm. In this context, multiple scenarios are designed and experimented in this research:

S1 Single Robotic Systems:

- **S1.1** Shaped-world scenarios
- **S1.2** Mapped-world scenarios
- **S1.3** Maze-world scenarios
- **S1.4** Distributed-world scenarios

S2 Multi-Robotic Systems:

- **S2.1** Search and rescue scenarios
  - **S2.1.1** Shaped-world scenarios
  - **S2.1.2** Maze-world scenarios
  - **S2.1.3** Distributed-world scenarios
- **S2.2** Search and coordinated rescue scenarios

5.1.1 Shaped-world Scenarios

The scenarios in this category include *alphabetical* situations like E, H, S, U, W, X shaped walls or various combinations of these shapes. Figure 5.1 presents one possible orientation of these shapes. Each shape offers a unique geometric situation to test the ability of a robot to cope with a certain "trap". A 'W' shaped scenario with one target, for example, is used to test the robot's ability to avoid prismatic traps by coming out of *increasingly difficult* situations, because the farther a robot moves into the valley, the harder it is to turn back. One such instance is presented in fig. 6.12(e). The following table 5.1 presents a list of shaped-world scenarios along with their utility in testing the IBF.

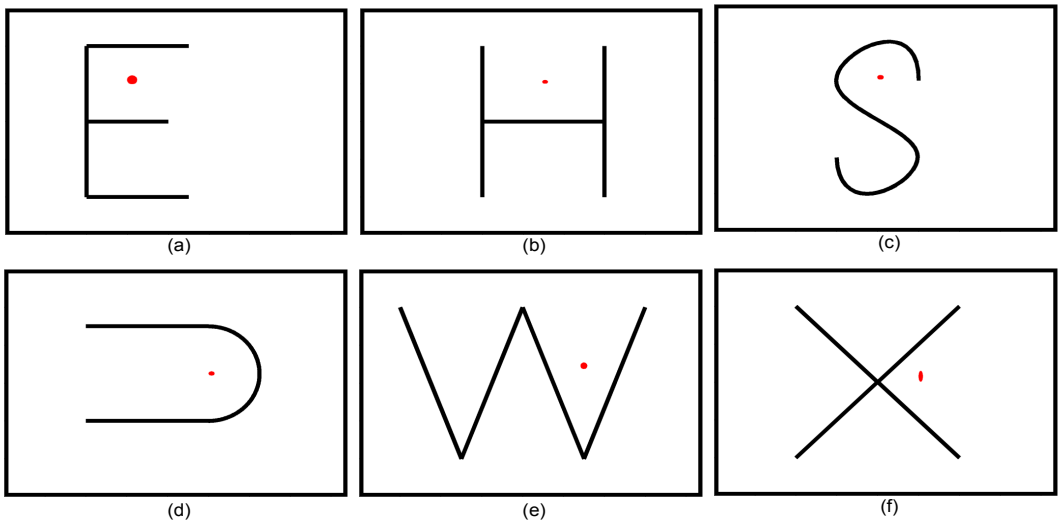


Figure 5.1: Different shaped-world scenarios

Table 5.1: Utility of shaped-world scenarios

Scenario		Application	Utility
S1.1	**E, H**	Single Robot	The ability of a robot to enter a rectangular space (room), albeit walls
			The ability of a robot to come out of a room and enter the other
S2.1.1		HMRS	Framework's ability to successfully navigate different robots
S1.1	**S**	Single Robot	The ability of a robot to enter a polar space
			The differences in searching clockwise or counter clockwise
S2.1.1		HMRS	Framework's ability to successfully navigate different robots
S1.1	**U**	Single Robot	Robot's ability to enter a narrow deep space
			The ability to turn around and attempt to enter again
S2.1.1		HMRS	Framework's ability to successfully navigate different robots
S1.1	**W, X**	Single Robot	Avoidance of a prismatic trap
			Performance in repetitive traps
S2.1.1		HMRS	Framework's ability to successfully navigate different robots

5.1.2 Mapped-world Scenarios

Mapped-world scenarios, on the other hand, are primarily employed to tabulate the performance metrics in order to offer a comparison with other well-known approaches. These is a library of different scenarios that researchers have used in their robotic applications and have a good citation index. These include scenarios of Borenstein's group [9, 10, 42, 78], Minguez

and Montano [57] and Fernández et al. [25]. Figure 5.2 shows the scenarios, whereas table 5.2 presents the utility of each scenario. Although, the scenarios are designed for single robot navigation, a comparative analysis can be performed to assess the navigation capability of immunity-based HMRS.

Table 5.2: Utility of mapped-world scenarios

Scenario		Application	Utility
S1.2	**Borenstein 1**	Single Robot	The ability of a robot to avoid obstacle and reach target The ability of a robot to resist the temptation of moving in free space
S1.2	**Borenstein 2**	Single Robot	Robot's capability to persistently move towards an intermittently visible target Avoidance of obstacles
S1.2	**Ulrich**	Single Robot	Robot's ability to avoid one obstacle and one trap in a corridor
S1.2	**Minguez**	Single Robot	Robot's ability to navigate through a premapped arena
S1.2	**Fernandez 1**	Single Robot	Robot's ability to pass through a narrow passage in a mapped world The ability of robot to avoid traps due to sparse nature of mapped data
S1.2	**Fernandez 2**	Single Robot	Robot's ability to navigate through multiple narrow passages The ability of robot to avoid traps due to sparse nature of mapped data

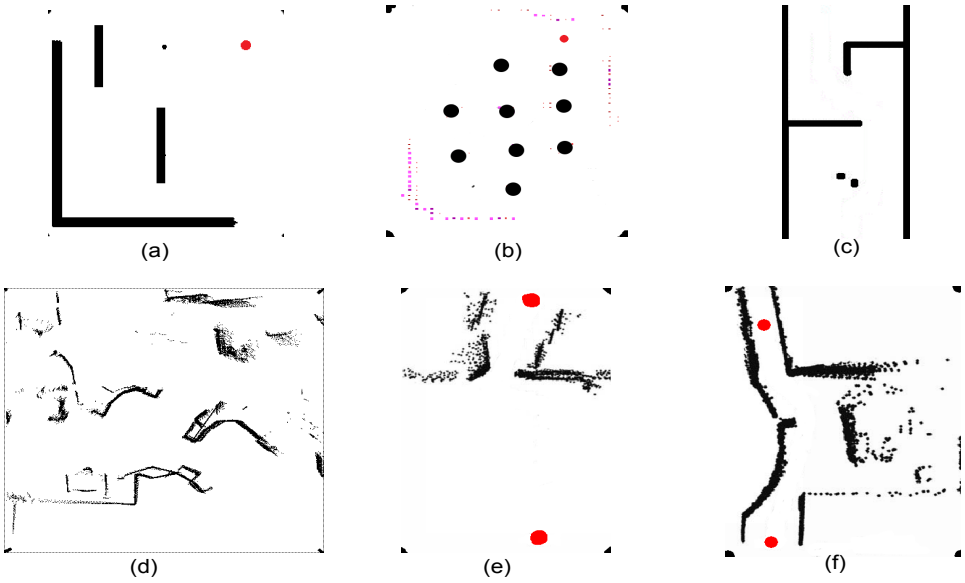


Figure 5.2: Different mapped-world scenarios: (a) & (b) from Borenstein et al. [10], (c) from Ulrich and Borenstein [78], (d) from Minguez and Montano [57] and (e) & (f) from Fernández et al. [25]

5.1.3 Maze-world Scenarios

Maze-world scenarios are primarily employed to establish that a robot can find its way out of a maze. Additionally, a successful navigation through a maze illustrates the robot's ability to avoid collisions, traps and cyclic behaviors. In this research, the scenarios are also scaled up to measure robot's performance in large mazes along with its learning ability from previous experiences. Figure 5.3 presents different maze scenarios with increasing complexity, whereas table 5.3 indicates their utilities in different applications. The passage clearing methodology is adapted from Meyer's maze game [55].

Table 5.3: Utility of maze-world scenarios

Scenario		Application	Utility
S1.3	**Maze**	Single Robot	The ability of a robot to navigate through a maze-world
			Robot's ability to avoid traps & collisions
			Robot's ability to avoid cyclic behavior
S2.1.2		HMRS	Framework's ability to successfully navigate different robots
			Framework's ability to guide follower-robots on the basis of learned experiences
S1.3	**Maze (scaled up)**	Single Robot	Online learning capability through robot-navigation
			Minimizing the analogous/cyclic behaviors
S2.1.2		HMRS	Framework's ability to successfully navigate different robots through large mazes
S1.3	**Maze with food**	Single Robot	Robot's ability to forage in addition to navigation
			Testing the reinforcement function of the framework
S2.1.2		HMRS	Robots' ability to successfully forage, albeit heterogeneity
S1.3	**Maze with Obstacles**	Single Robot	Robot's ability to avoid obstacles in addition to the walls of maze
S2.1.2		HMRS	Framework's ability to successfully navigate different robot's while avoiding obstacles

5.1.4 Distributed-world Scenarios

The arenas in distributed-world scenarios are implemented to establish the effectiveness of IBF in unstructured environment, with increasing task densities. The sizes and coordinates of obstacles/targets are randomly gener-

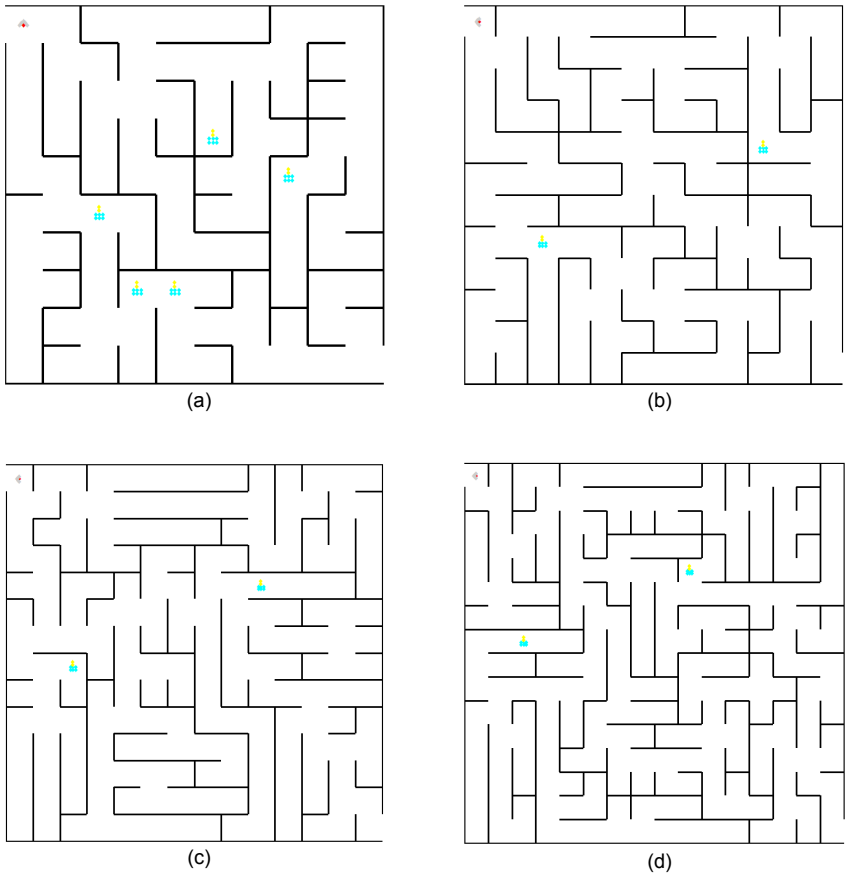


Figure 5.3: Different maze-world scenarios

ated, while avoiding overlaps. This creates a sparse distribution of obstacles and targets in the arena, as shown in Fig. 5.6. Alternatively, the user also has the freedom to place obstacles/targets on her own. In addition to the robot heterogeneity, fig. 5.6 also indicates different starting positions for each experiment. Figure 5.4 presents different scenarios with increasing complexity, whereas table 5.4 indicates their corresponding utilities.

Table 5.4: Utility of distributed-world scenarios

Scenario		Application	Utility
S1.4	**Randomly distributed targets and obstacles**	Single Robot	The ability of a robot to navigate through randomly distributed targets and obstacles
			Effectively search targets with various starting positions
			Performance evaluation w.r.t increasing task density
S2.1.3		HMRS	Framework's ability to successfully navigate different robots through the obstacles
			Framework's ability to learn from previous experiences (online)
			Effectively search and rescue targets with various starting positions
			Performance evaluation w.r.t increasing task density
S1.4	**User-defined distribution of targets and obstacles**	Single Robot	Robot's ability to reach difficult/hidden corners of the arena for targets
			Robot's ability to come out of difficult corners of the arena
			Performance in conflicting goals
S2.1.3		HMRS	Framework's ability to guide different robots through difficult situations

In addition to aforementioned scenarios, *predator-prey* and *search-and-coordinated-rescue* experiments are also conducted. The task complexity in predator-prey experiments do not require communication between different robots whereas others do. Coordinated rescue, on the other hand, even requires the transfer of navigation strategy. The detailed results of all the experiments are presented in next chapter.

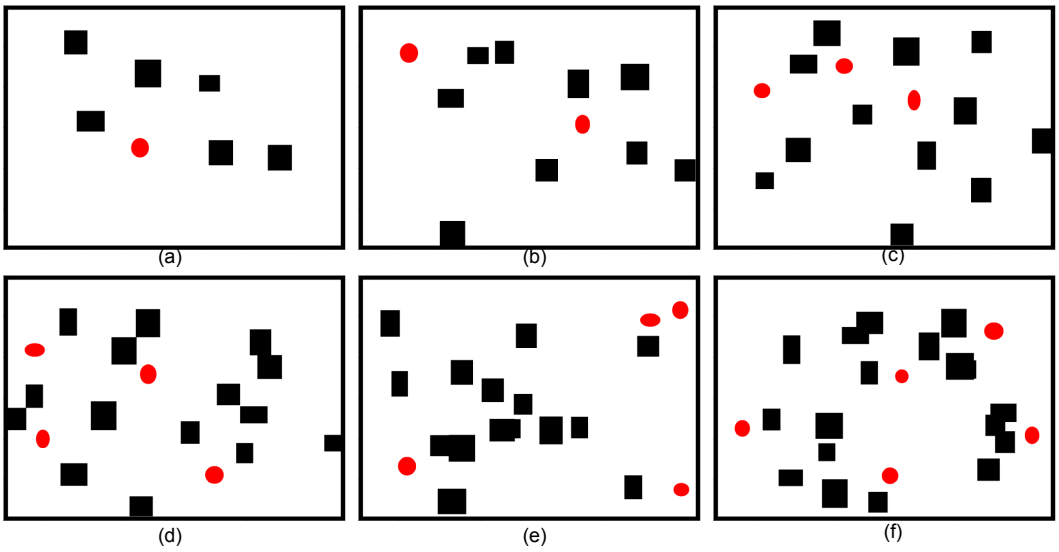


Figure 5.4: Different distributed-world scenarios

5.2 Robot configurations

A library of pre-configured robots with different sensors (e.g. camera, infrared/photosensors, SICK and sonar) and actuators (e.g. grippers of different jaw-length) is maintained in the simulator. Moreover, a user is offered to pick different sensors/actuators to configure her/his own robot. Robot platforms, holonomic or non-holonomic, are also incorporated. Alternatively, the user has the flexibility to choose a simpler representation, if required. The underlying immune system, which is the heart and soul of simulator, has already been described in Chapters 3 and 4.

Robot configurations for different experiments are tabulated in table 5.5. It is evident that each robot is different from other robots in terms of its morphological details. Sensory specifications of search robots have two

components: one to navigate with the help of localizing sensors and second to detect the targets by using *speciality* sensors. Rescue robots, on the other hand, have additional actuatory capability to pick the targets to be rescued, according to their sizes. The sensory configuration is described in terms of the range of sensing e.g. a robot with photosensor-configuration of $\{[-\pi/3, \pi/3]$, $30\}$ indicates that it can sense from $-\pi/3$ to $\pi/3$ at 30 equi-spaced instances, if robot's azimuth is centered at 0 `[rad]`. Table 5.5 also lists the unique identification code (R_i) for each robot. It later serves as a part of antibody structure which is stimulated by clonally evolved idiotopes.

Table 5.5: Robot configurations used in SAR experimentation

Robot	**Role**	**R_i**	**Sensors**		**Actuators**
			Localization	Speciality	
Robot A	Search	{0 0 0 1}	Camera View-field: $2\pi/3$	None	None
Robot B	Search	{0 0 1 0}	Laser range finder Sense-angle: $4\pi/3$	Color	None
Robot C	Search	{0 0 1 1}	Sonar Cone-angle: $4\pi/5$	Color	None
Robot D	Search	{0 1 0 0}	Photosensor $[-2\pi/3, 2\pi/3]$, 30	Color	None
Robot E	Rescue	{1 0 0 1}	Photosensor $[-\pi/2, \pi/2]$, 20	None	Gripper Jaw-length: 30`[mm]`
Robot F	Rescue	{1 0 1 0}	Photosensor $[-\pi/3, \pi/3]$, 30	None	Gripper Jaw-length: 40`[mm]`
Robot G	Rescue	{1 0 1 1}	Photosensor $[-2\pi/5, 2\pi/5]$, 35	None	Gripper Jaw-length: 40`[mm]`

The antigen mapping schematic is illustrated in fig. 5.5. It illustrates that a single situation is mapped by different robots in terms of different

lengths of epitope structure of the antigen (environment). The epitope, in turn, stimulates clonally evolved paratopes of antibodies according to the strength of match. The resulting network-dynamics from stimulations and suppressions output the antibody concentrations (steering directions). The antibody concentration rates, as specified by hybrid representations in eq. 4.10, automatically adjusts to different string lengths on the basis of R_i.

5.3 Evaluation

Evaluation of the framework is done on multiple levels in order to establish its effectiveness, using different scenarios. The first level is to test the underlying immune system. BIS has an internal mechanism to evaluate its performance in terms of inflammation and memory. Successful execution of IBF, therefore, reduces the inflammation levels to maintain a homeostasis. Immune-memory is another tool to judge the evolved immune system. The memory-bank starts empty and ends up with a collection of evolved B/T lymphocytes, with higher affinities towards antigenic data, indicating that cloning and maturation is successful.

The second level is to evaluate the performance of a robotic system is to use cost, utility and reliability metrics. Cost metrics includes the *computational cost and time* along with the corresponding variance. Utility metrics are defined in terms of *heterogeneity* and *scalability*. Robot heterogeneity is implemented by systematically changing the sensor types and locations (as in table 5.5) and measuring the consequent performance metrics. Similarly,

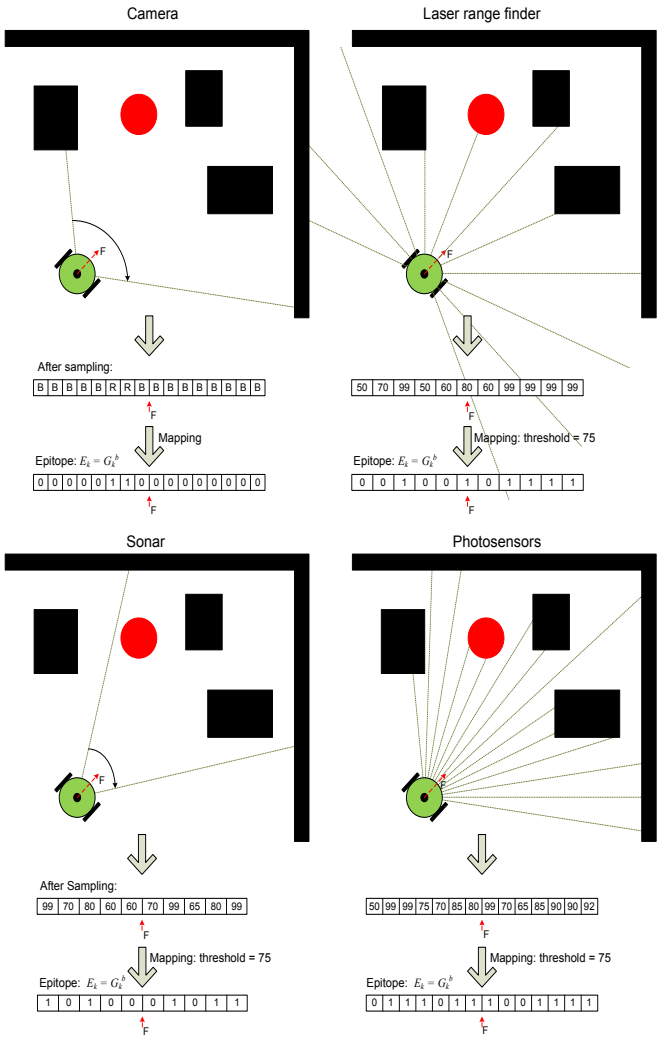


Figure 5.5: Antigen mapping with different robot morphologies

scalability of proposed framework is tested by measuring performance metrics by gradually increasing the task-density, number of robots and/or robot behaviors.

Reliability metrics include *collision-count* and *mission-success* corre-

sponding to increasing obstacle/target density. It is expected that collision-occurrence will reduce with time because of negative selection during T-cell maturity and consequent maintenance of immune-memory. Sensitivity to different parameters in underlying mathematical expressions is also analyzed. Moreover, *mission-success* (MS) is recorded using scoring criterion of annual *Robotic Rescue* competition [40]. This serves as a benchmark for SAR applications.

$$\mathsf{MS} = 50 \cdot \left\{ \frac{m/M + 1 - t/T}{H^2} \right\} \tag{5.1}$$

where, H corresponds to the number of human interactions, t is the task completion time of the total allotted T, m is the number of successfully detected targets out of a total of M. The criterion ensures that a score of 50 to 100 is possible if all the targets are rescued with an assurance that the only human interaction is that of starting the simulation.

At the third level of evaluation, the aforementioned performance metrics are computed using conventional idiotypic network approach and compared with those of the framework. In Chapter 6, the results corresponding to these evaluation criteria are presented and discussed in detail.

5.4 Simulator

The simulator is implemented in MATLAB. It has all the aforementioned functions of scenario specifications, robot-configurations and perfor-

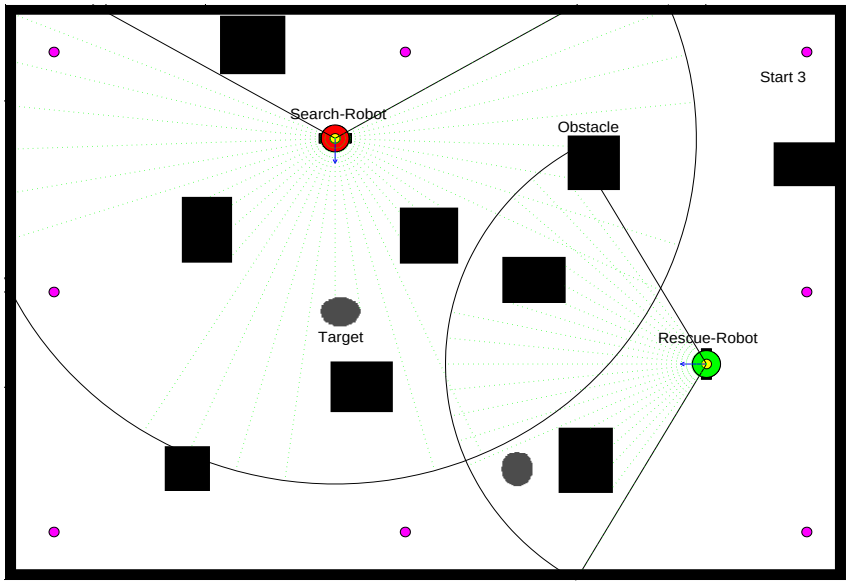


Figure 5.6: One instance of simulation involving two robots with different sensing capabilities. Different starting positions are also shown in a distributed-world scenario.

mance evaluation, in addition to the core functionality of the AIS. Moreover, it supports multiple starting positions, arena displays, ODE solvers, plotting utilities and transformations. A user interface also enables a start/stop option during a simulation run. The traces of robots' positions, orientations, collisions, energy levels and immunity levels are recorded for future references as well. Figure 5.6 shows one instance of IBF execution with labels indicating starting positions, robots' morphological details and arena configurations. Similarly, fig. 5.7 provides the user interface for an HMRS experiment. In the instance shown, the inflammation has dropped as consequence of a successful

rescue.

5.5 Chapter Summary

The chapter is focused on experimentation details of IBF and starts by discussing multiple scenarios to test its validity. The pros and cons of each subset of scenario-configuration is considered. Multiple tables are tabulated along with their utilities. It is followed by providing the particulars of robot-configurations for SAR application. The morphological details of each robot and corresponding mapping to antigen is provided in support. Evaluation criteria is laid out including metrics related to immunological working of the framework as well as the robotic system. The chapter is concluded by providing the brief of the simulator functions.

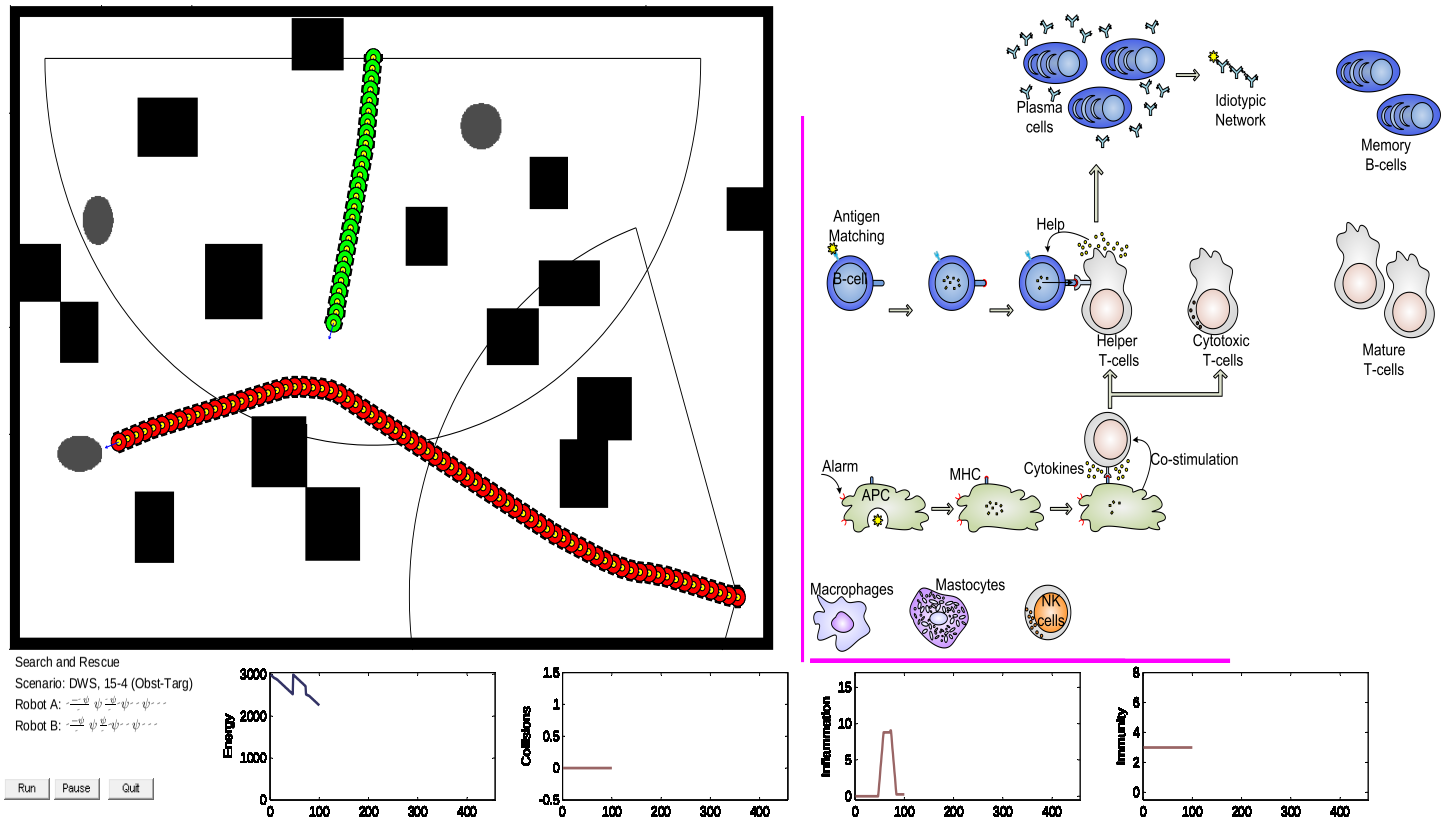


Figure 5.7: User interface of the simulator output. Shown are the virtual environment (left), graphs for Energy, Collisions, Inflammation, and Immunity. The pink bars at x and y axis indicate the maturity of immune-response and its depth, respectively. In the virtual environment, traces of where the robots (2 in this case) have been.

Chapter 6

Results and Discussion

Extensive simulations with different robot-configurations, scenarios and starting positions are conducted, according to the previously defined experiment specifications, to validate the framework on multiple levels. Each subcategory includes a brief description of the simulation-runs, the results and a discussion on the findings. Multiple screenshots of the experiments are also included to augment the discussion.

6.1 Single-Robotic Systems

The simulation results involving single-robotic systems are included to provide a comparative analysis with other immunological robotic applications.

6.1.1 SAR in a mapped-world

Mapped-world scenarios were simulated with different robot morphologies to exhibit the algorithm's ability to handle an SAR task in various situations. Figure 6.1 and associated results in table 6.4 illustrate that each robot can search an initially hidden target, navigate through narrow passages, avoid obstacles and rescue the target by following the IBF. Figures 6.1 (a), (b), (c)

and (d) show the simulations in obstacle-courses designed by Borenstein and Koren [10]. Similarly, sub-figures (e), (f), (g) and (h) show the capability of presented methodology to navigate through narrow passages and maps of Fernandez, et al. [25]. The scenarios of Fernandez, et al. are a result of a SLAM algorithm and therefore, contain regions of poorly mapped areas. Each screenshot also corresponds to a different robot-morphology. Each scenario was tested with all the robot-configurations of table 5.5, irrespective of their role in a SAR application. The selection of these scenarios is done on the basis of MoVeME benchmarking system of Cailisi and Nardi [14].

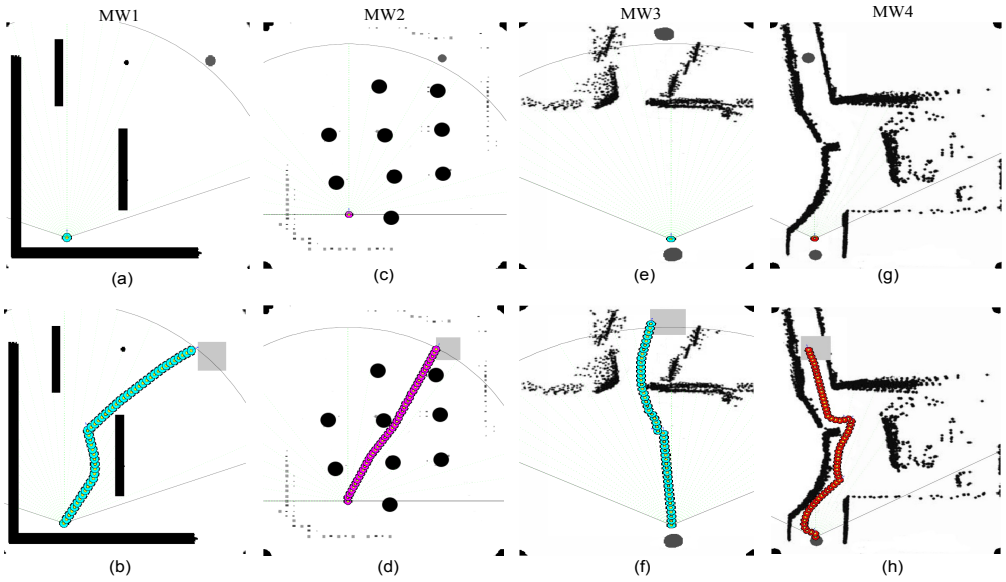


Figure 6.1: Simulations in mapped-world scenarios with different robot-morphologies

The results in table 6.1 refer to the statistical findings of repeated experiments. Each experiment was repeated with all the robot morphologies in

Table 6.1: SAR results in mapped world scenarios

Scenario	Statistic	Time (s)	Event count EC	Collision Count CC	Energy Level EL
MW1	**Mean**	34	47	0	2586
	Std. dev.	19	1		9
MW2	**Mean**	67	45	1	2664
	Std. dev.	35	0		47
MW3	**Mean**	41	59	11	2536
	Std. dev.	18	31		200
MW4	**Mean**	103	58	2	2548
	Std. dev.	68	4		41

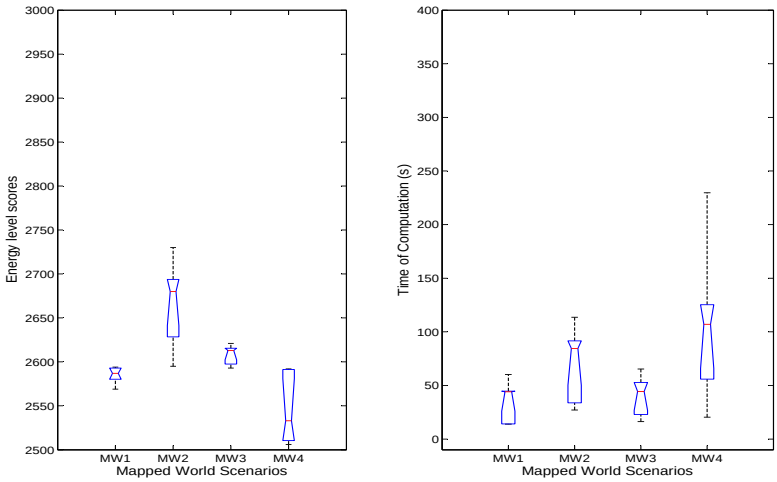


Figure 6.2: Performance evaluation in mapped-world scenarios

table 5.5. The cumulative collision-count in scenario-MW3 is found to be high. The major contributor in this collision count is robot-configuration-A which experienced a total of 5 collisions. It is because of a the sampled viewfield of robot's camera. A total of 20 samples of camera's viewfield makes it difficult to detect small-distant-target(s) and the robot more prone to the accidents. Al-

ternatively, the number of samples of sensor reading can be increased because the algorithm supports variable lengths of data. The variance in energy level is also high in scenario-MW4 because robot-B failed to to enter the narrow passage in its first attempt but succeeded in the second. It can be concluded that the IBF is able to successfully navigate different robots of table 5.5.

6.1.2 Navigation through a maze

The maze-world scenarios are employed to test the limits of IBF in terms of its ability to navigate through difficult situations. A robot does not have any knowledge of its world outside its sensing range (one *unit* in four directions). It only can take one step in either direction, if open. It is the task of IBF to help the robot to come out of the maze by finding a single open path, from start to finish. The different mazes are randomly generated by specifying the number of rows and columns. Moreover, the maze-sizes are gradually increased to test the scalability of IBF in terms of arena difficulty. An increase of two rows and columns corresponds to a total increase of 44 units with respect to a 10x10 maze. A total of 4 increasingly difficult maze-configurations were selected for experimentation, including 10x10, 12x12, 14x14 and 16x16 mazes. Each configuration was used to generate 5 random maze-patterns. Each maze-pattern was then experimented thrice to test the IBF.

A robot treats each instance of its sensory information as bacteria and moves to *kill* it according to phagocytosis code defined in Chapter 3. It also secretes monokines as a trail of its movement. Once a robot encounters a

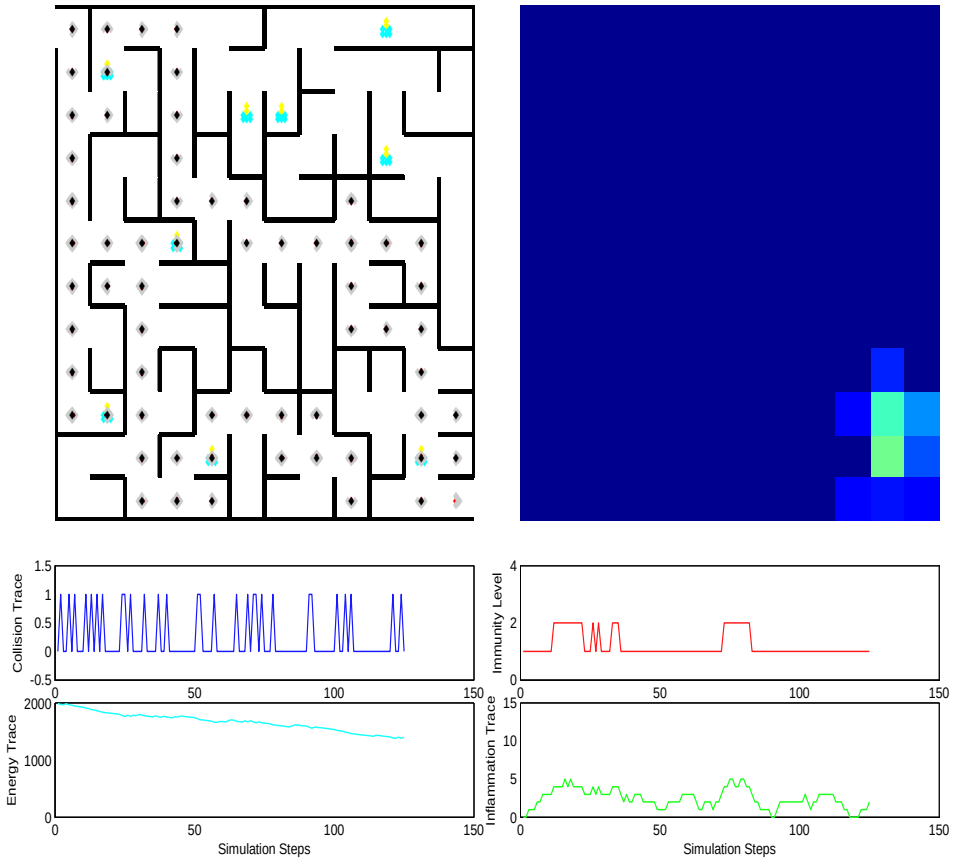


Figure 6.3: Navigation through a 12x12 maze with performance indicators

recent-repeat situation (e.g. coming out of a closed path) it decides on the basis of diffused gradient of its monokines to avoid revisiting the same trap. It is important to highlight that chemoattractants/chemorepellents diffuse and, consequently, a *distant-repeat* situation can make a robot revisit the trap. Figure 6.3 shows one simulation-run in a 12x12 maze along with the traces of collision count, energy levels, immunity levels and inflammation. The results show that the immunity level jumps to the adaptive layer if inflammation

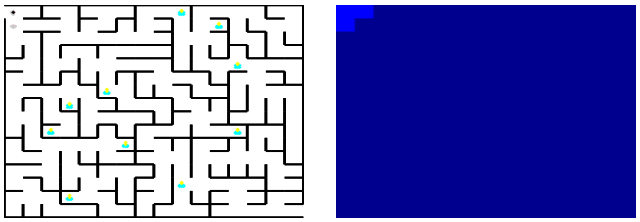
rises in the case of the innate layer being unable to contain it. A collision is inevitable in case of a three-walled trap because of robot's inability to *wait* according to Rodney Meyer's movement-definition [55]. A repellent trail of monokines forces the robot to bump once before it is able to turn around.

Figure 6.4 illustrates the different stages of a robot's movement in a 16x16 maze. A zoom in–and–out of the figure shows the movement from start to finish. Similarly, Fig. 6.5 presents different simulations in increasing maze-sizes. The results are presented in table 6.2. It shows a success rate (MS_1) of 87% in the cases of 10x10 and 12x12 mazes. The success rate decreases with the increase in maze-size but it is a consequence of limiting the time of experimentation. A 16x16 maze with $t_F = 400$ indicates an increase in MS_1 scores. Figure 6.6 indicates performance indicators of time of computation and mission-success scores (MS_2) in aforementioned mazes.

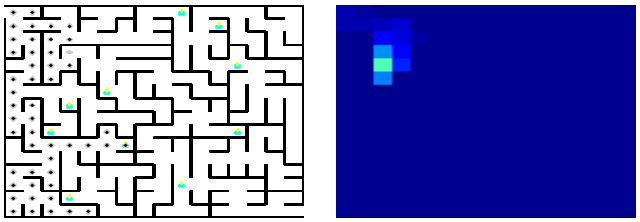
$$MS_1 = \begin{cases} 1, & \text{if } \mathbf{p} == \text{size}(\texttt{maze}); \\ 0, & \text{otherwise.} \end{cases} \tag{6.1}$$

$$MS_2 = 50 \cdot \left\{ m + 1 - \frac{t}{250} \right\} \tag{6.2}$$

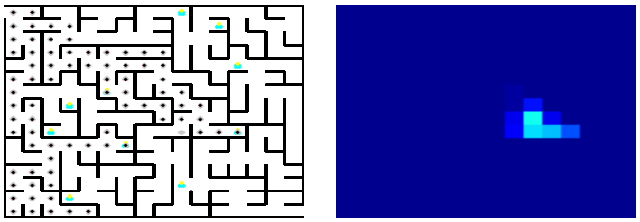
It can, therefore, be concluded that the IBF is able to navigate a robot through difficult maze-scenarios by incorporating a reactive approach to move a robot, a diffusing monokine-secretion to avoid a trap and an adaptive layer to reduce the inflammation levels.



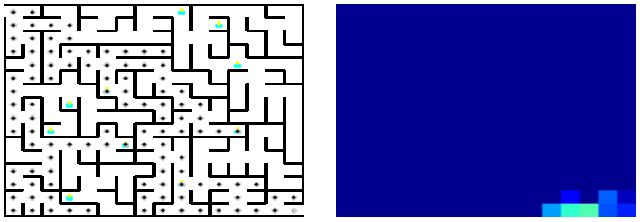
(a)



(b)

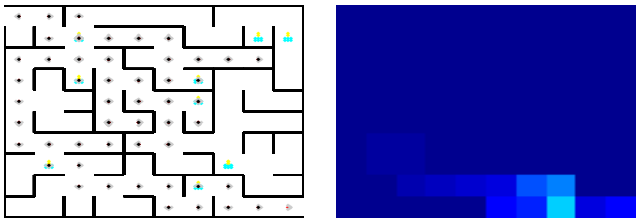


(c)

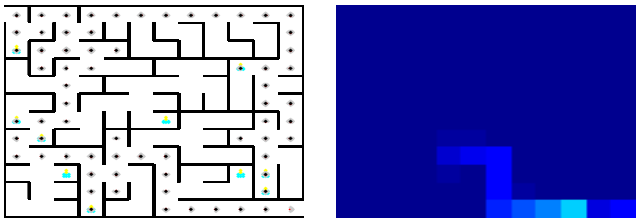


(d)

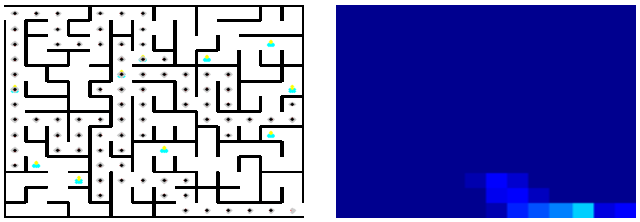
Figure 6.4: Different stages of robot's movement through a 16x16 maze.



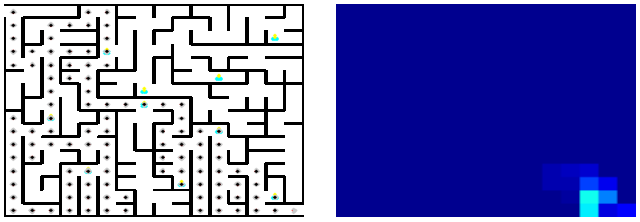
(a)



(b)



(c)



(d)

Figure 6.5: Illustration of framework's ability to navigate through increasingly difficult mazes.

Table 6.2: Results of robot navigation through maze-world scenarios

Maze	Statistic	Time (s)	Event count EC	Collision count CC	Inflamm'n (acc.) AI	Mission Success MS1 (%)	MS2
10x10	**Mean**	5.378	126	36	381	87	79
	Std. dev.	2.691	62	20	224		8
12x12	**Mean**	9.665	159	45	476	87	62
	Std. dev.	4.008	61	20	221		29
14x14	**Mean**	12.724	177	52	551	67	49
	Std. dev.	4.895	69	22	244		35
16x16	**Mean**	18.238	216	71	760	53	33
	Std. dev.	3.679	42	13	148		33
16x16	**Mean**	21.118	250	82	871	80	56
($t_F = 400$)	**Std. dev.**	6.929	84	26	290		31

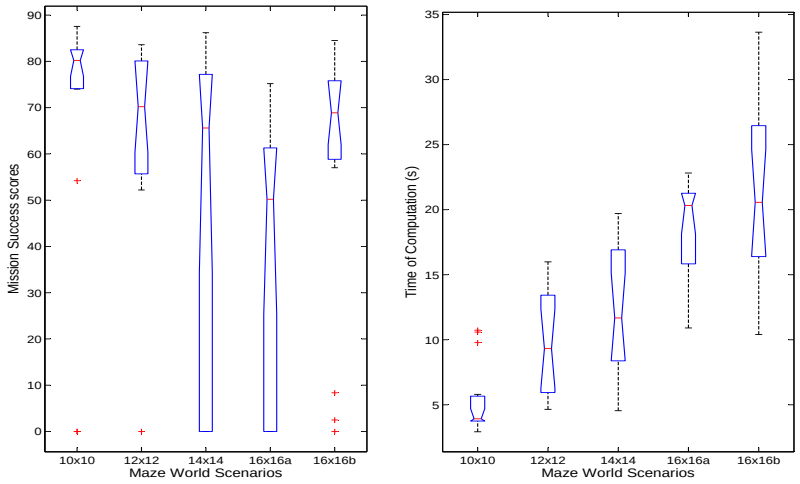


Figure 6.6: Performance evaluation in maze-world scenarios

6.1.3 Single robot in a distributed-world

In a distributed-world, different robot morphologies, according to table 5.5, were simulated in increasing obstacle-target densities. Figure 6.7(a) shows a robot with $\left\{[-\pi/3, \pi/3],\ 16,\ \pi/24\right\}$ configuration, all set to embark upon

an SAR mission. Figure 6.7(b) indicates that the robot started with a monocytic navigation, exploring the arena while avoiding obstacles. Once a target was sighted, the inflammation started rising which consequently triggered an adaptive-immunological response to rescue it. A successful rescue, illustrated as a light-grey patch, reduces the inflammation as well as the immunological response to a T-lymphocyte level. The robot soon found another target which was rescued at a B-lymphocytic level with clonal selection as well as idiotypic network functionalities. After a second successful rescue the inflammation level reduces to another minima and so does the immunological response of the IBF. The last target is also rescued in a similar fashion.

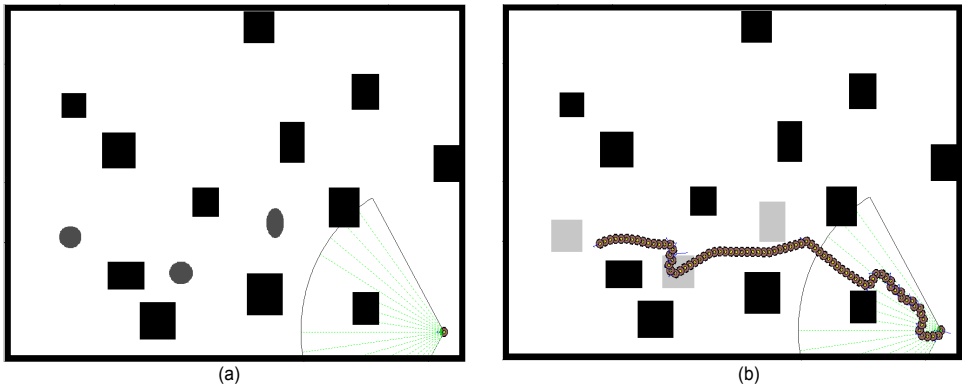


Figure 6.7: One robot searches three targets in a distributed-world scenario

A collision-free navigation, as indicated in Fig. 6.8, and a successful rescue of all the targets in the arena indicates IBF's ability to avoid obstacles and seek targets, respectively. Figure 6.9, on the other hand, illustrates that the framework is also able to handle different starting positions for an SAR

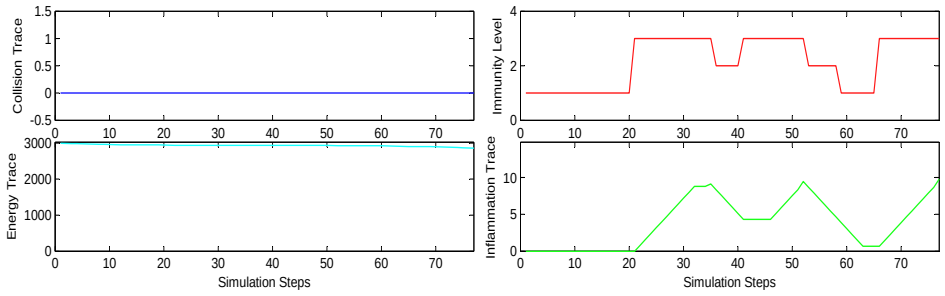


Figure 6.8: Performance indicators corresponding to single-robotic navigation in a distributed-world scenario

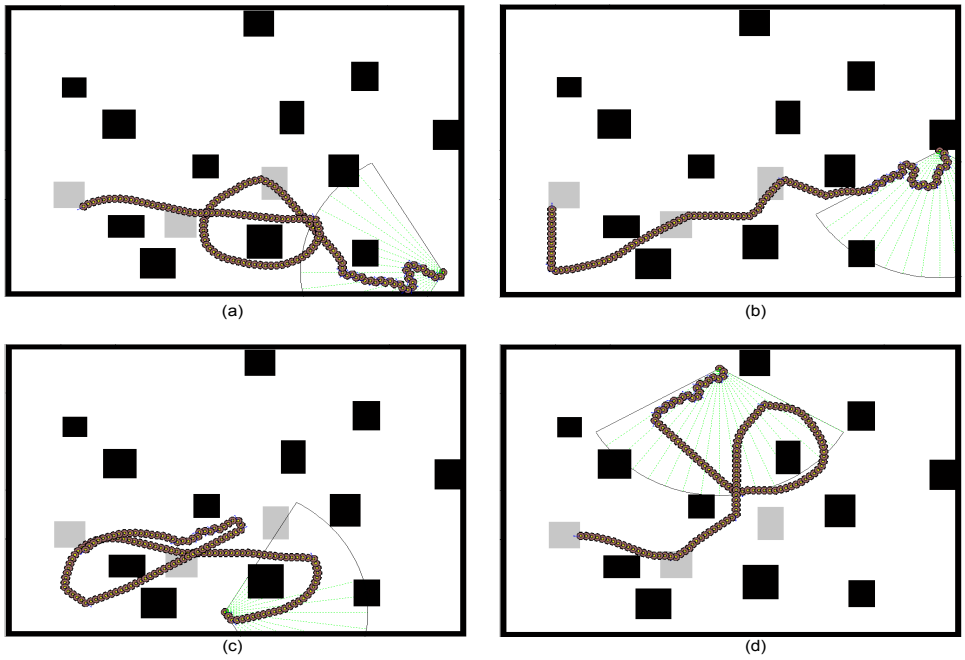


Figure 6.9: SAR with single robot with different starting positions in a 12-3 Obstacle-target arena configuration

task with a single robot.

6.2 Heterogeneous Mobile Robotic Systems

The simulation results for systems involving heterogeneous robots are presented in the following subsections to establish the validity of IBF in a multitude of different scenarios, robot-configurations and experiment specifications.

6.2.1 HMRS in a shaped-world

Different stages of a two-robot SAR task are presented in Fig. 6.10. It shows that the search-robot starts searching the target(s) in the arena (8.a). Each instance of sensor reading is treated as bacteria which disturbs the homeostasis, invoking the stimulation of monocytes as steering agents following a probability density estimation on the basis of the present gradient of chemoattractant/chemorepellent molecules. Once a target is sighted, adaptive functions are invoked and antibody-concentrations are regulated by plasma cells and consequent idiotypic network of antibodies that in turn navigate the search-robot towards the target, as shown in Fig. 6.10(b). The search-robot in the vicinity of a target communicates the target location to the rescue robot by transmitting the communication-antigen (8.c). This antigen disturbs the homeostasis of the team of rescue robots. The rescue robot with highest affinity towards the presented communication-antigen is triggered to move towards the target (8.d). In addition to the robot-heterogeneity, it is important to highlight that the rescue-robot, while moving towards the target, avoids the collisions with walls/obstacles. Figure 6.10(e) also shows the probabilistic

nature of the robot locomotion, where the robot seems to keep going and retraces its steps to reengage in the pursue of the target. Once the rescue robot reaches close to the pick-up range of gripper, the target is tagged as *rescued*, as shown in Fig. 6.10.f, and results in reducing the inflammation and dendritic cell maturity.

A total of eight different starting positions for each scenario were tried, as shown in Fig. 6.11, to test the algorithm's ability to successfully conduct an SAR task. An E-shaped scenario is shown with different starting positions for a search robot, according to the specifications in Fig. 5.6. It is clear that the algorithm has an ability to navigate robots towards the targets, irrespective of their starting positions and morphologies. Similarly, other shaped scenarios are also tested and their sample simulation runs are illustrated in Fig. 6.12.

The results in table 6.3 refer to experiments with shaped-world scenarios where each experiment was repeated with different starting positions. The table lists the aforementioned performance indicators for all of the scenarios. It is worth noting that the robots take longer in the case of W shaped scenarios. It is because of the combined stimulative-suppressive effect of $\cos(\Delta\theta_{ij})$, which ensures that steering directions in the neighborhood of frontal direction are stimulated more, forcing the robot to move into a valley. Moving into a valley is important because it ensures exploration of previously unknown regions in a map but it is equally important to move out of it. The hybrid Idiotypic network ensures it by adding another antigen in addition to previous

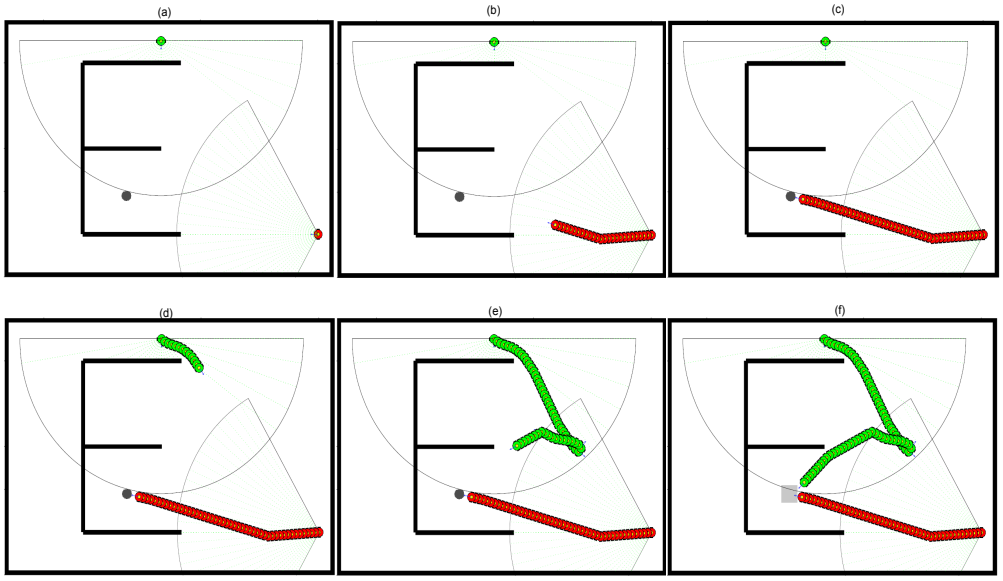


Figure 6.10: Shaped-world scenario (E) with different stages in a simulation run

antigens in n_{ik}. In the case of U-shaped scenarios as well, a longer time span is indicated in table 6.3 because once a search robot misses the only opening of a horizontal U, it has to cover all the arena again to possibly find an entrance into the shape. It is important to highlight that unlike path planning, a robot in this setup does not know the existence of a target nor its location, *a-priori*. It is, therefore, important for a robot to move into prismatic-valleys (shapes: W & X), rectangular-traps (shapes: E & H) and circular-traps (shapes: S & U), in order to find possibly elusive target(s) as well as to move out of those traps, possibly unscathed. Mission-success scores indicate that the robots are successful in terms of finding the targets in distress and rescuing them, albeit different starting positions and robot morphologies. The statistical results of

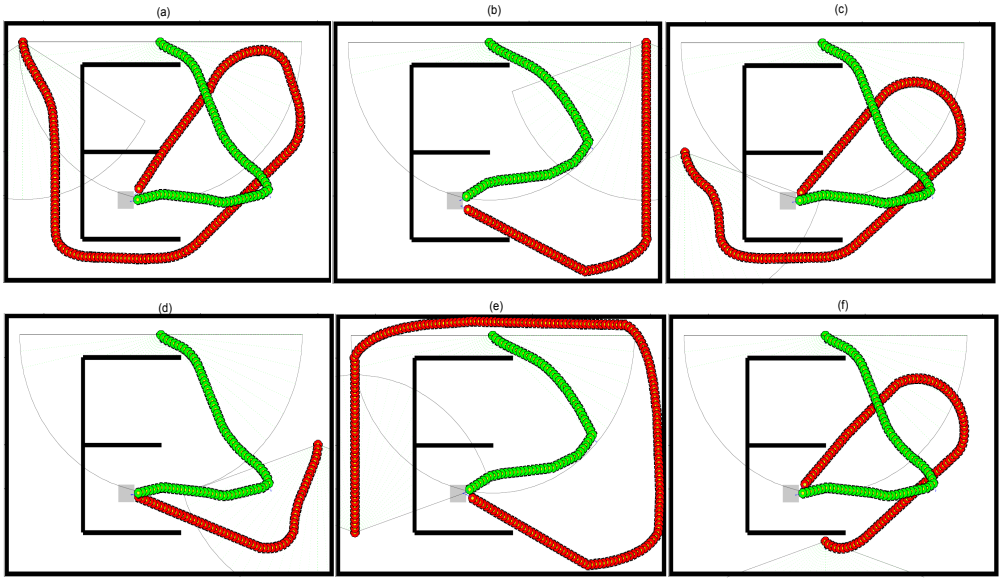


Figure 6.11: Multiple simulations in a shaped-world scenario (E) with different starting positions

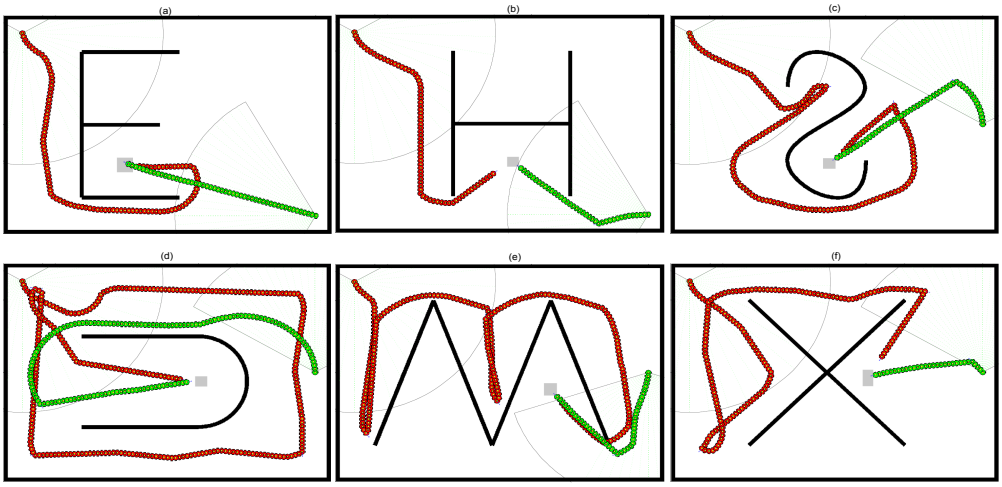


Figure 6.12: Different shaped-world scenarios for a two-robot SAR task

these simulations in terms of *time* and MS are also presented as box-plots in Fig. 6.13.

Table 6.3: SAR results in shaped world scenarios

Scenario	Statistic	Time (s)	Event count EC	Collision Count CC	Mission Success MS
E	**Mean**	72	157	0	92
	Std. dev.	29	59		3
H	**Mean**	109	132	16	93
	Std. dev.	65	48		2
S	**Mean**	125	166	0	92
	Std. dev.	104	56		3
U	**Mean**	169	366	0	82
	Std. dev.	55	115		6
W	**Mean**	288	194	12	90
	Std. dev.	271	120		6
X	**Mean**	102	147	0	93
	Std. dev.	100	69		3

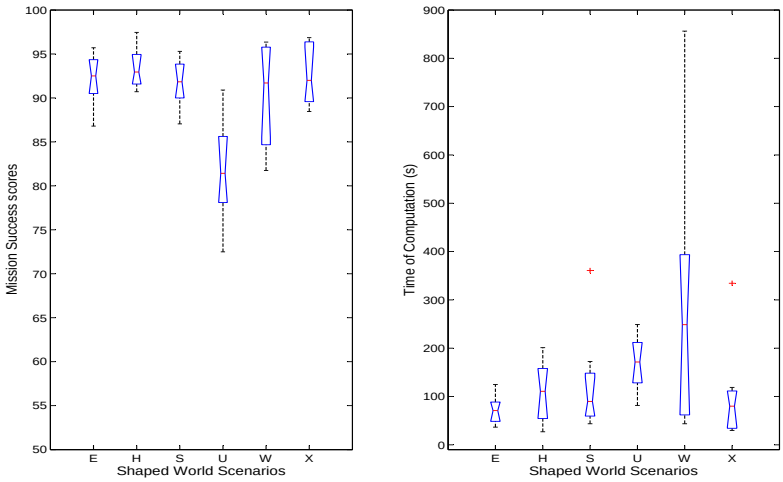


Figure 6.13: Performance evaluation in shaped-world scenarios

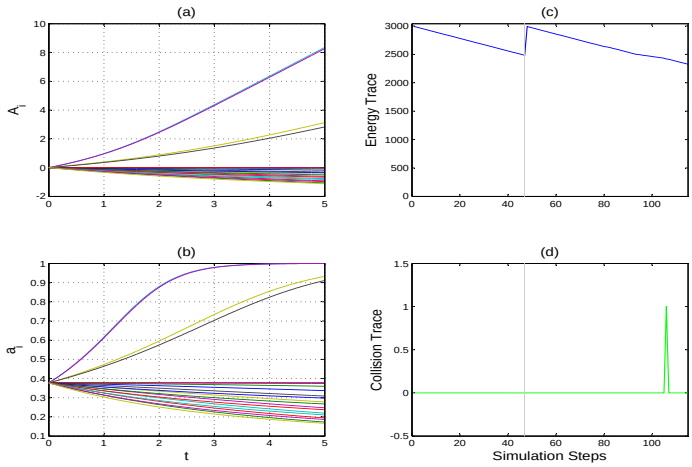


Figure 6.14: Performance indicators during one simulation run: (a). Antibody concentrations in one iteration with zero initial conditions, (b). Squashing function for antibody concentrations, (c). Energy trace, (d). Collision trace.

Figure 6.14 shows the results of antibody concentrations, according to eq. 4.10 with zero initial conditions. It illustrates, in sub-figure (a), that some antibodies have increased in concentrations as a consequence of the presented antigens at that time instance, whereas others have been suppressed because of the idiotypic network. The antibodies with the highest concentrations are selected as an elitist selection to steer the robot in the respective direction, according to the representation scheme in eq. 4.2. The squashing function, as illustrated in sub-figure 6.14(b), ensures the antibody concentrations to be within $[0, 1]$. Energy and collision traces are shown in sub-figures (c) and (d) to visually indicate the respective data with respect to simulation steps. The

grey vertical line indicates the switch from search to rescue behaviors.

6.2.2 HMRS in a distributed-world

Multiple simulations in distributed-world scenarios are presented in figures 6.15 and 6.16. Figure 6.15 (a) and (b) corresponds to a 6-1 obstacles-target configuration at the start and finish of the simulation run, respectively. Adjacent sub-figures to the right correspond to gradual increase number of obstacles and targets in the arena. It shows that the algorithm is able to *search-and-rescue* even if the obstacle/target density is increased. Figure 6.16, on the other hand, provides snapshots of two-robot SAR simulations in a 15-4 obstacle-target configuration. Sub-figures (e) to (f) refer to instances of each rescue whereas sub-figures (b) to (d) show results with different robot morphologies with different starting positions.

The results in table 6.4 refer to the experiments with distributed-world scenarios where each experiment was repeated with different starting positions. Moreover, each scenario is denser than the previous one in terms of its obstacle/target density. The experiments are successful because the underlying AIS is able to *search-and-rescue* all the targets. Some collisions are experienced in scenarios with higher number of obstacles and targets e.g. DW3 and DW4. These are primarily caused by lower number of samples of sensed data, as indicated in the discussion for mapped-world experimentation. It is, there-

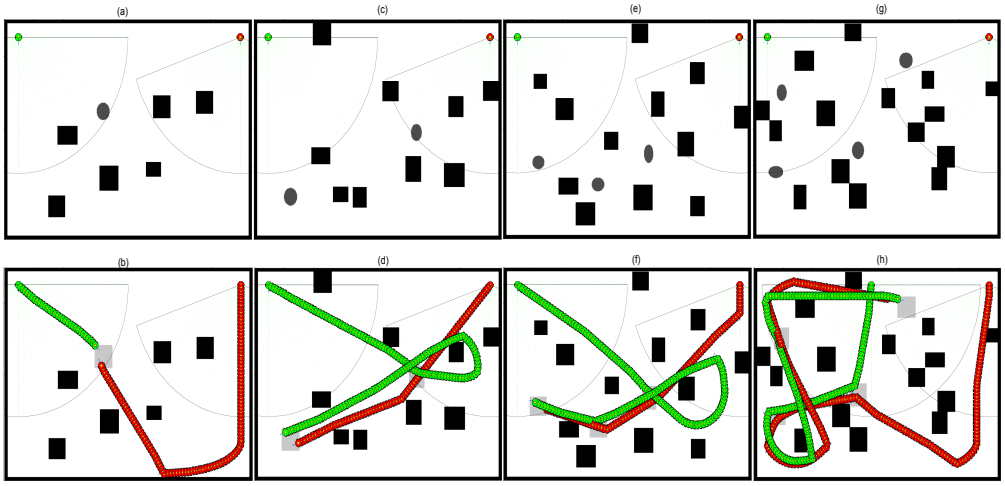


Figure 6.15: Simulation runs with increasing obstacle/target density in distributed-world scenarios

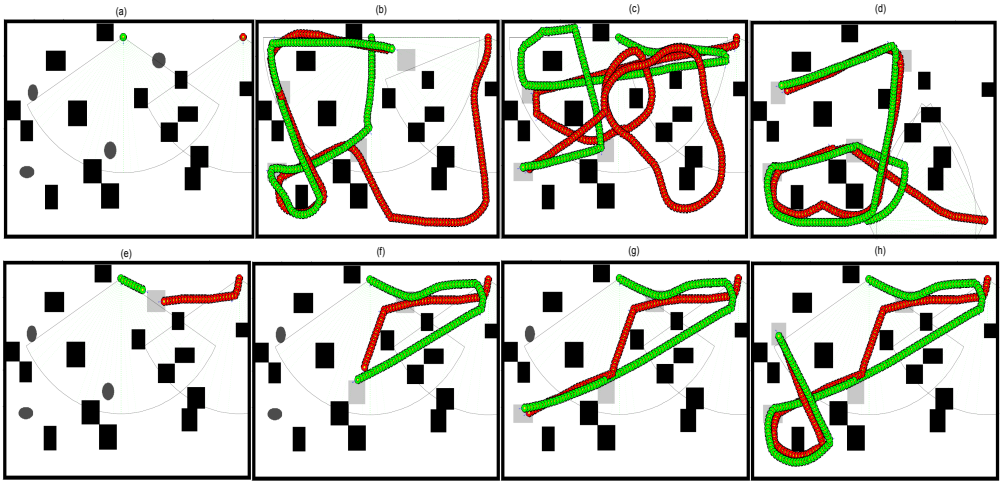


Figure 6.16: Simulation screenshots of various distributed-world scenarios and starting conditions

fore, suggested that the number of samples in antigenic data be increased, whenever situations are denser and collision-prone. All the simulations have

good MS-scores that signify that search-robots can find all the targets even if obstacle/target density is increased and similarly, rescue-robots can find their way to the targets to pick them up, irrespective of the differences in sensory/actuatory capabilities. Figure 6.17 gives statistical information of *time* and *mission-success* data in terms of box-plots of experiments in each scenario.

Table 6.4: SAR results in distributed world scenarios

Scenario	Statistic	Time (s)	Event count EC	Collision Count CC	Mission Success MS
DW1	**Mean**	51	66	0	97
	Std. dev.	22	29		1
DW2	**Mean**	175	208	3	91
	Std. dev.	54	72		3
DW3	**Mean**	143	169	19	93
	Std. dev.	56	62		3
DW4	**Mean**	260	433	21	82
	Std. dev.	120	140		6

In the context of the aforementioned successful executions of multiple SAR scenarios, it can be claimed that the IBF is capable to handle robot-heterogeneity as well as the navigation in unstructured environments. This claim is further substantiated with the subsequent results, in tables 6.1, 6.2, 6.3, and 6.4.

6.3 Computational Environment

All the simulations were executed using MATLAB, on a system with a Intel® Core™i7-2675QM CPU @2.20`[GHz]` with 6.00`[GB]` RAM. The tabulated *time* parameter is computed using MATLAB's own `tic`, `toc` functions.

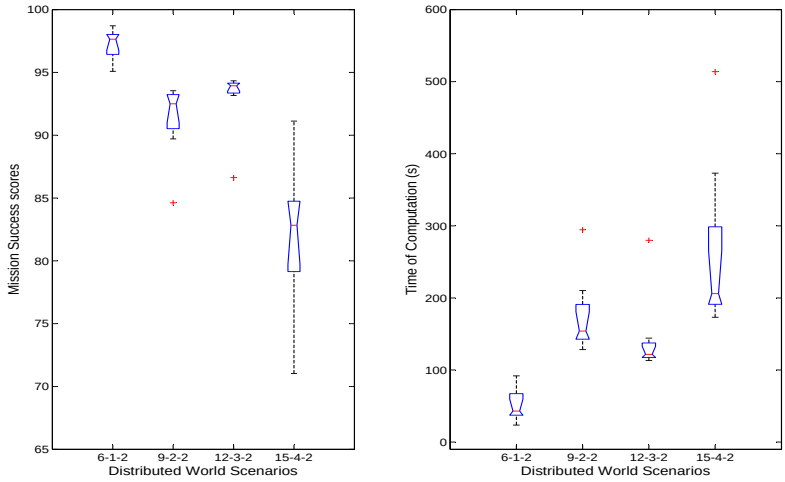


Figure 6.17: Performance evaluation in distributed-world scenarios

The *event-count* is the number of the times immune-functions are invoked, whereas the *collision-count* is calculated as the combined total of all the collisions. Moreover, the metric of *mission-success* is computed using eq. 5.1 of Robotics Rescue competition [40].

6.4 Comparison with Other Robotic Approaches

In order to compare the performance of the IBF developed with other approaches in robotics, two representative techniques are selected. The first technique is the *vector field histogram* (VFH) approach of Borenstein and Koren [10], whereas the second methodology, the called *reactive immune network* (RIN), is a recent immunologically-inspired approach of Luh and Liu [50]. The selection of the VFH approach was motivated by the fact that the MoVeME

benchmarking mechanism uses it as a reference. Moreover, a good volume of literature is available for comparison. The selection of the recent approach of Luh and Liu is done to signify the scope of IBF, vis-à-vis state of the art in immunity-based robotics. One of the most important aspects of framework's capabilities is its flexibility in terms of successfully executing SAR tasks on a wide spectrum of scenario-sets, albeit morphologically different robots. Other approaches in robotics, on the other hand, focus on specific applications without considering system's flexibility outside the similar environments or different robots.

The vector field histogram (VFH) methodology was reconstructed according to details in the literature and simulations were conducted using the same machine with identical configurations of simulator, robots, arenas and starting conditions. The results of the robotic SAR task, using the VFH methodology, are illustrated in Fig. 6.18. It shows the poor performance of VFH algorithm in valleys. Once a comparative observation is made between the results of IBF (Fig. 6.12) and VFH (6.18), the superior performance of IBF is established in shaped-world scenarios especially in terms of its capability to come out of narrow valleys and avoid deterministic or cyclic navigation. It is important to highlight that the improvements of the basic VFH method require additional specifications of behaviors, cost functions, heuristics and/or algorithms to minimize the cost functions [78]. The IBF, on the other hand, is based on both the reactive and adaptive functionalities of an immune system and their inter-connectivities. The VFH approach and its variants can work

in the cases of open spaces e.g. situations in which the arena boundary is not defined and, therefore, having a limited scope for practical applications.

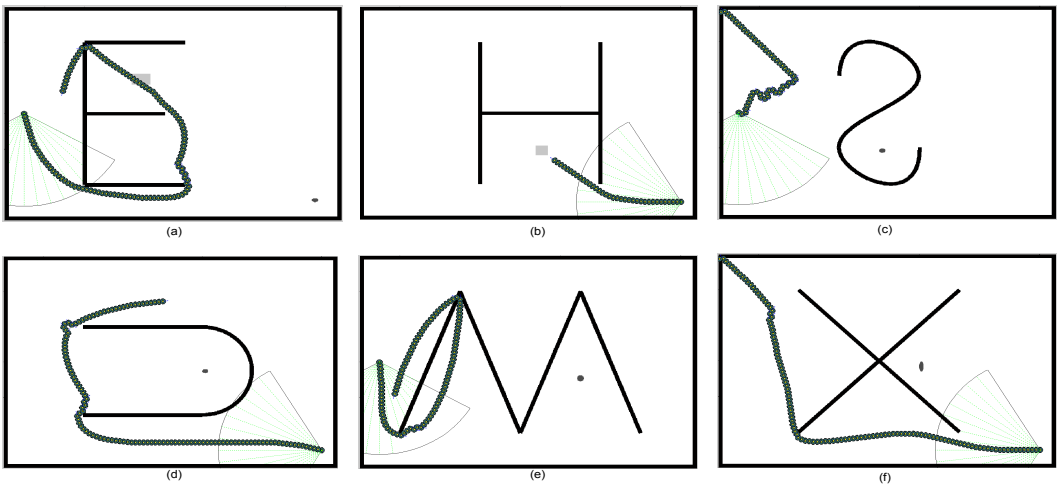


Figure 6.18: Performance of VFH-method in different shaped-world scenarios indicating its limitations w.r.t the performance of IBF in Fig. 6.12

A comparison of the reactive immune network (RIN) of Luh and Liu with the framework is presented in Fig. 6.19 for maze-world scenarios. It shows only one instance of successful navigation through a maze of 10x10. A mission success rate of only 7% is observed according to the experiment specifications of §6.1.2. It is important to highlight that RIN also needs an additional specification of target location contrary to a *no-target-a-priori* approach of IBF. The IBF not only solves the maze, its performance in scaled-up arenas is already established in table 6.2. The limitation of RIN primarily arises from its inherent bias towards the direction of predefined target location. The only maze solved by RIN also indicate that the open path from start to finish was

in-line with the direction of target location, as viewed from the start location. Although a virtual target method [91] is incorporated in the RIN application, it does not work well in cases of narrow passages in the arena.

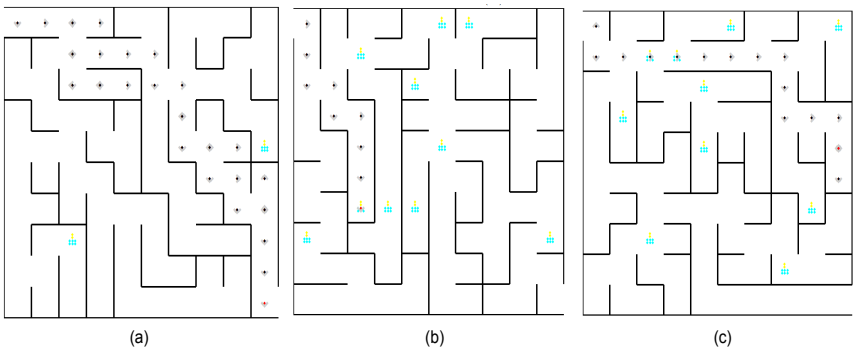


Figure 6.19: Performance of RIN [50] in a maze world indicating its only success in a 10x10 maze

Table 6.5: Comparison between IBF and Reactive Immune Network of Luh and Liu [50]

Maze Size	Statistic	Time (s)	Event count EC	Collision Count CC	Mission Success MS1	MS2
The IBF						
10x10	**Mean**	5.378	126	36	87	79
	Std. dev.	2.691	62	20		8
RIN of Luh and Liu [50]						
10x10	**Mean**	25.142	218	116	7	17
	Std. dev.	4.962	52	35		20

6.5 Chapter Summary

This chapter presents a detailed account of simulations to establish the performance of the framework. It is structured in three sections. The first

section presents and discusses the simulation-results of single robotic applications on various scenarios including mapped, mazed and distributed arenas. The second section extends the discussion to include heterogeneous systems with shaped and randomly distributed arenas. The third section presents additional comparison with other robotic approaches. The experimented scenarios are also gradually scaled up e.g. distributed-world scenarios are started from a 6-obstacle–1-target arena and gradually scaled up to 15-obstacles–4-targets in the same dimensions of the arena.

In light of the extensive experimentation, mentioned above, and the detailed discussion, it is concluded that the IBF can successfully be applied on a multitude of scenarios, ranging from maze-arenas to environments with randomly distributed obstacles/targets. Moreover, it can handle heterogenous robotic systems using a three-tier immunological algorithm while performing the behaviors of wander, obstacle avoidance, target seeking and rescuing.

Chapter 7

Conclusions and Future Directions

7.1 Conclusions

The dissertation research shows that an immunity-based framework, combining the functions of innate and adaptive immunities, is not only possible but successful as well. It is designed as a framework that offers both holistic and distributed approaches towards heterogeneous mobile robotic applications. The holistic nature is exhibited by a display of morphologically different robots performing a coordinated search-and-rescue task. The distributed nature, on the other hand, is reflected through independent navigation of individual robots.

The methodology designed in this research is novel in terms of abstracting the functions of innate immunity into an HMRS. A collection of virtual steering agents (monocytes), housed within the robot(s), move in a biased-random manner by following a Monte Carlo methodology, towards the mapped sensory data (bacteria). The resulting probability density of *virtual* steering agents guides the robot to move. The robots' experiences, in a window of recent past, increase or decrease the internal health indicator (inflammation). Once the robot(s) move, a virtual trace of their movement is left to diffuse in

the arena. A follower robot may encounter the diffusing trace of the leading robot as a positive reinforcement (chemoattractant) to continue-on or as a negative reinforcement (chemorepellent) to avoid the path. Another module is that of contextualizing the sensed environment on the basis of a window of recent experiences, both internal and external. This is achieved through the functionality of dendritic cells. In a nutshell, all the robots in the arena move independently, experience good or bad situations and consequently contextualize the environment for all the robots in the arena, if innate immunity module is active.

If a situation is found to be persistently *dangerous* the immunity level jumps to the adaptive layer. It is important to highlight that the representation schema designed for the framework is compatible across the immunological levels. For each robot, the monocytes, dendritic cells, T and B lymphocytes are equal in number, if a cross-tier functionality is invoked. The length of representation, however, can be different for each robot according to its morphological configuration. It means that if a robot has 30 monocytes on basis of 30 sensors around its periphery, it will have the equal number of dendritic, T and B cells. But the number can be different for other robots in the HMRS.

The results indicated that the inflammation level rises, if recent experiences include a collision or a persistent target, and consequently helps in regulating the responses of the IBF. The regulatory role of inflammation is only confined to arbitrate the levels of immunological response. Additional factors, specific to a particular application, may also be included to increase or

decrease the inflammation e.g. a successful rescue in an SAR scenario reduces it.

On the second tier of IBF, T-cells are called to perform the positive and negative selections on the basis of energy and collisions, respectively. The immunity level jumps again if T-cells are unable to resolve the situation. The B-cell layer responds by evolving the *actions* through cloning and hyper-mutation. This is followed by the network dynamics to output the concentrations of actions (antibodies). The antibodies with higher concentrations are selected as the actuation signals for each robot in the system. Moreover, the previous traces of a robot's movement continue to diffuse and consequently help in avoiding future occurrences of bad behaviors.

The IBF's performance, in the context of successful experimentation, clearly illustrates that it is capable to handle the tough situations, robot-heterogeneity and navigation in unstructured environments. The comparisons of the frameworks' performance with that of VFH [10] and RIN [50] approaches also substantiate the claim of the framework's effectiveness. It is also concluded that the IBF is flexible in terms of executing the SAR tasks on a wide spectrum of scenario-sets, albeit the morphologically-different robots. This claim is further strengthened by eliminating the need of predefined behavior modules through the regulatory role of inflammation and dendritic cell maturity. It is important to highlight that the IBF consistently performed the assigned tasks, even in the cases of increased complexities of maze-world and randomly-distributed scenarios.

The research outcome suggests that the immunological inspirations can be deeper and researchers should benefit from a multitude of different cells and their specific functionalities, along with their mutual interactions. A shift from conventional B-cell-approach or B-T-approach is possible to structure a multi-tiered or a cyclic approach involving a multiple immune-functions.

7.2 Future Directions

The framework has a great potential for future expansions, both in terms of computational immunology and robotics. The IBF currently employs an immune-memory to contain successful actions and a regulatory mechanism of inflammation and DC-maturity to gradually improve its responses, over a period time. It is also possible to add a feedback from selected/successful antibodies to the future activity of monocytes. In biological immunity a similar mechanism exists. Moreover, the dissertation research does not differentiate between immune-functions of the blood stream or the tissue. Future modifications can include such distinction if an application poses a demand.

A number of possible robotic applications can be implemented and tested on the developed framework e.g. swarm intelligence on a team of homogeneous robots, predator-prey simulation, multi-robot box-pushing experiments, progressive enhancement of robotic platforms, robotic hardware evolution, graceful degradation of robotic systems, etc. An interesting extension of this research could be to test the effects of vaccinations on *sick* robots.

The framework's library of sensors, actuators and kinematic platforms

can further be enhanced to include popular models of mobile robots. It is also possible to include physical health indicators e.g. infra-red thermometer to monitor processor's temperature as a factor in the inflammation module. Similarly, the accelerometers can be used to physically detect bumps.

Appendices

Appendix A

Details of Immuno-inspired Robotic Applications

Table A.1: Robotic Applications using Clonal Selection Theory

	Experiment Specifications	Auxiliary Functions
Hu [30, 31]	Robot path planning,	Roulette wheel selection, Mutation, insertion & deletion operators
Wang & Hirsbrunner [83]	IMEA for robot path planning, no hypermutation or cloning	GA: crossover & mutation
Li et al. [48]	Single robot performing concurrent mapping & localization	GA: crossover Vaccination operator
Hur [33]	Multi robot bomb disposal system with three robots (Scanner, inspector, diffuser)	Memory up-dation, Cloning with $N_c = \sum_{i=1}^{n} round(\frac{\beta.N}{i})$
		Table ended

Table A.2: Mathematical Details of CS-based Robotic Applications

	Antigen (G_k)	Antibody (A_i)	Affinity
Hu [30, 31]	—	Grid points of Robot path, $X = [a_o, a_1, \cdots, a_n]$ a_o = start pt, a_n = end pt	$fitness = \frac{A}{f}$, $f = \sum_{i=1}^{N} (d_i + c\beta_i)$ $d_i = \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}$ $\beta_i = \sum_{j=1}^{M} \alpha_j$ (if obstacle, 0 otherwise)
Wang & Hirsbrunner [83]	Problem	Possible solution	$F_i = \nu_i \left[1 + \zeta \frac{\nu_i}{max(\nu)} + \delta D_s \left(1 - \frac{\nu_i}{max(\nu)}\right)\right]$ $D_s = \frac{\sum_i (\theta.max(\nu) \leq \nu_i \leq max(\nu))}{\frac{\mu + \phi}{M.N}}$ $\nu_i = \frac{}{\sum_i^N \sqrt{(x_{pi} - x_{qi})^2 + (y_{pi} - y_{qi})^2}}$
Li et al. [48]	—	Modeled as chromosomes $X = [X_1, X_2, \cdots, X_N]$ where $X_j = [\Delta d_j, \Delta\theta_j]$	$f = \sum min(1 - O_{ij}, 1 - E_{ij} + w_1 \sum \delta_{ij} + w_2 \sum \zeta_{ij})$ $O_{ij} \rightarrow$ occupancy $E_{ij} \rightarrow$ empty
Hur [33]	Binary coded $\{0,1\}^2$	Binary coded $\{0,1\}^3$, Action	$f = \frac{1}{\left[1 + \sum_i^k w_i.n_i\right]}$ $k \rightarrow$ no. of soft constraints $n_i \rightarrow$ no. of constraints within an A_b
			Table ended

Table A.3: Robotic Applications using Idiotypic Network Theory

	Experiment Specs	A_i Selection	Cloning/Metadyn.	Auxiliary Functions
Ishiguro et al. [34, 36]	Single robot: seek energy and avoid obstacles	Roulette wheel	—	Reinforcement learning to compute T_x
Watanabe et al. [85]	Single robot: collect garbage, seek energy and avoid obstacles	Roulette wheel	M: GA-based	Mixing pot crossover function
Michelan & Von Zuben [56]	Single robot: collect garbage, seek energy and avoid obstacles	Roulette wheel	—	GA-based adjustment scheme with elitist selection
Vargas et al. [79, 80]	Single robot: collect garbage, seek energy and avoid obstacles	Roulette wheel	—	GA-based adjustment and learning classifier system
Krautmacher & Dilger [43]	Single robot: simple rescue scenario	—	—	Bilinear transformation for network dynamics
Wang et al. [84]	Single robot: path planning, seek goal and avoid obstacles	—	—	Strengthened learning (RL), Obstacle restriction method
Tsankova et al. [75]	Stigmergy-based foraging behavior with two networks	Roulette wheel	—	*a-priori* antibody idiotopes and paratopes
Whitbrook et al. [86, 89]	Robot navigation in maze world: with fixed idiotope and adjustable paratope matrices	Feedback	C: A_i concentrations M: GA-based	RL: $P[x_w, y_d](t+1) = max(0, P[x_w, y_d](t) + rf_{t+0.5})$
Lee and Sim [46]	Multi-robot system: homogeneous, mine detection	Winner takes all	C: Strategy transfer	FIS with task density as input
Jun et al. [39]	Multi-robot system: homogeneous, mine detection	T-cell assisted	C: Strategy transfer	T-cell concentration: $\eta(1-g_i)A_i$
Li and Wang [47]	Dog-sheep system: homogeneous robots	Synthesized IN	—	T-cell: $\eta(1-g_i)$
Duan et al. [23]	Predator-prey system: homogeneous robots	Synthesized IN	—	—
Luh et al. [51]	Robot soccer: homogeneous robots	Winner takes all	—	FIS for m_{cw}, m_r and m_b

continued on next page

Table A.3 – continued from previous page

	Experiment Specs	A_i Selection	Cloning/Metadyn.	Auxiliary Functions
Luh and Liu [50]	Single robot: reactive navigation	Winner takes all	—	Virtual target method for local minima recovery
Dehuai et al. [22]	Single robot (car like): path planning	$a_i = (1-\gamma_i)a_i^o + \gamma a_i^g$ γ through distance	—	—
				Table ended

Table A.4: Representation Scheme of IN-based Robotic Applications

	Antigen (G_k)	Antibody (A_i)	Affinity
Ishiguro et al. [34, 36]	Pre-massaged Data [object, direction, energy level] | Binary coded epitope	Behavior modules (fixed), Binary coded paratope (Desirable Action) | Action | Idiotope (ID of disallowed A_b)	Adjustment Mechanism
Watanabe et al. [85]	Binary coded, pre-massaged [object, direction, energy level] | Binary coded epitope	Behavior modules (fixed), Binary coded paratope | Action | Binary coded Idiotope	Hamming distance
Michelan & Von Zuben [56]	Binary coded, pre-massaged [object, direction, energy level] | Binary coded epitope	Behavior modules (fixed), Binary coded paratope | Action | Stimulated A_b (idiotope)	Hamming distance
Vargas et al. [79, 80]	Ternary, through classifier [object, direction, energy level] | Tagged, {0,1,#} epitope	Behavior modules (2 part paratope), Antecedent Paratope | Consequent paratope | Stimulated A_b (idiotope) Tag: Ternary, {0,1,#} | Binary encoded	Hamming distance
Krautmacher & Dilger [43]	Binary encoded, variable Object Type (fixed): {human,wall,empty} | Obj Position (variable): 12 sectors | binary encoded (epitope)	Behavior modules, Binary coded paratope | Action | Binary coded: Idiotope | Idiotope	Hamming distance
Wang et al. [84]	Binary encoded, pre-massaged, [obstacle, near/far] | Epitope-obstacle [target, near/far] | Epitope-target	Binary coded (separate) Paratope-obstacle | Action $[(1-r_i).a^o_i + r_i.a^g_i]$ | Idiotope-obstacle Paratope-target | Idiotope-target	Hamming distance
Tsankova et al. [75]	Binary coded, pre-massaged [object, direction, energy level] | Binary coded epitope	Behavior modules (fixed), Binary coded paratope | Action | Binary coded Idiotope	Hamming distance
Whitbrook et al. [86, 89]	8 predefined situations [Situation, priority] | Antigen array G(x)	16 predefined actions, [Action, Speed] | Paratope matrices (P_x) Idiotope matrices (I_x)	Strength of match by reinforcement scores
Lee and Sim [46]	Real, classification of task density Task density [High, Medium, Low, None]	Real, action strategy, Behavior: [Aggregation, Random search, Dispersion, Homing]	Combined stimulation and suppression
Jun et al. [39]	Real, classification of task density Task density [High, Medium, Low, None]	Real, action strategy, Behavior: [Aggregation, Random search, Dispersion, Homing]	Combined stimulation and suppression

continued on next page

Table A.4 – continued from previous page

	Antigen (G_k)	Antibody (A_i)	Affinity
Li and Wang [47]	Real, Environment coding X = $(x_1, x_2, \ldots, x_n)$ x_i=[dog's position, sheep's position]	Real, Action strategy, Y = $(y_1, y_2, \ldots, y_n)$ y_i=[dog's position, sheep's position, Action]	String matching
Duan et al. [23]	Real, Environment coding X = $(x_1, x_2, \ldots, x_n)$, x_i = [predator's position, prey's position] CA = coordination antigen	Separate action strategies, Y = $(y_1, y_2, \ldots, y_n)$ y_i=[predator's position, prey's position, Action]	String matching
Luh et al. [51]	Real data A_g = {$distance_{ball\text{-}goal}$, $distance_{ball\text{-}robot}$, crowd}	6 predefined behaviors, A_{bi} = {kick, pass, chase, track, guard, shoot}	Average from FIS in a 6x6 matrix
Luh and Liu [50]	Real fused data, A_g = {θ_g, d_i, θ_{si}} Azimuth of goal position, sensor data (distance & location)	Steering directions, $A_{bi} \equiv \theta_i = \frac{360^\circ}{N_{Ab}}(i-1)$	—
Dehuai et al. [22]	Real, Task density [High, Low, None]	Real, Steering directions, Condition 1: $dist_{r\text{-}t}$, $\theta_{r\text{-}t}$ / Condition 2: $dist_{r\text{-}o}$, $\theta_{r\text{-}o}$ / Actions (8 directions)	—
			Table ended

Table A.5: Mathematical Details of IN-based Robotic Applications

	Suppression $(A_i - A_j)$	Stimulus 1 $(A_j - A_i)$	Stimulus 2 $(G_k - A_i)$
Ishiguro et al. [34, 36]	$\frac{T_p^{Ab_i} + T_r^{Ab_j}}{T_{Ab_j}^{Ab_i}}$	$\frac{T_r^{Ab_j} + T_p^{Ab_i}}{T_{Ab_i}^{Ab_j}}$	Hamming distance
Watanabe et al. [85]	$F\left[\sum_{k=1}^{L}(w_k\overline{I_i(k) \oplus P_j(k)})\right]$	$F\left[\sum_{k=1}^{L}(w_k\overline{I_j(k) \oplus P_i(k)})\right]$	$\sum_j\left[F\left(\sum_{k=1}^{L}(w_k\overline{E_j(k) \oplus P_i(k)})\right)\right]$
Michelan & Von Zuben [56]	$m_{ij} = \frac{T_p^{Ab_i} + T_r^{Ab_j}}{T_{Ab_j}^{Ab_i}}$	$m_{ji} = \frac{T_r^{Ab_j} + T_p^{Ab_i}}{T_{Ab_i}^{Ab_j}}$	Hamming distance
Vargas et al. [79, 80]	$F\left[\sum_{k=1}^{L}(w_k\overline{I_i(k) \oplus P_j(k)})\right]$	$F\left[\sum_{k=1}^{L}(w_k\overline{I_j(k) \oplus P_i(k)})\right]$	$\sum_j\left[F\left(\sum_{k=1}^{L}(w_k\overline{E_j(k) \oplus P_i(k)})\right)\right]$
Krautmacher and Dilger [43]	$F\left[\sum_{k=1}^{L}(w_k\overline{I_i(k) \oplus P_j(k)})\right]$	$F\left[\sum_{k=1}^{L}(w_k\overline{I_j(k) \oplus P_i(k)})\right]$	Hamming distance $\forall A_g$
Wang et al. [84]	$\sum_{k=1}^{L}(I_i(k) \oplus \overline{P_j(k)})$	$\sum_{k=1}^{L}(I_j(k) \oplus \overline{P_i(k)})$	$\sum_{k=1}^{L}(E_j(k) \oplus \overline{P_i(k)})$
Tsankova et al. [75]	$\sum_{k=1}^{L}(I_i(k) \oplus \overline{P_j(k)})$	$\sum_{k=1}^{L}(I_j(k) \oplus \overline{P_i(k)})$	$\sum_{k=1}^{L}(E_j(k) \oplus \overline{P_i(k)})$
Whitbrook et al. [86, 89]	$P[x_{w1}, y_m]I[x_i, y_m]H_i$	$(1 - P[x_i, y_p])I[x_{w1}, y_p]H_i$	$P[x_i, y_j]G(x_i)_j$
Lee and Sim [46]	Pre-computed γ_{ij}	Pre-computed γ_{ij}	g_i from FIS
Jun et al. [39]	Pre-computed m_{ij}	Pre-computed m_{ij}	g_i from FIS
Li and Wang [47]	$\sum_{k=1}^{N} m_{ik}g_k$	$\sum_{j=1}^{N} m_{ij}g_j$	Affinity: g_j
Duan et al. [23]	—	$\sum_{j=1}^{N} m_{ij}g_{ij_k}$	g_j & CA_{ik}
Luh et al. [51]	—	—	$\frac{m_{cw} + m_r + m_b}{3}$
Luh and Liu [50]	$\cos(\theta_j - \theta_i)$	$\cos(\theta_i - \theta_j)$	$f_{target} + f_{obstacle}$

continued on next page

Table A.5 – continued from previous page

	Suppression $(A_i - A_j)$	Stimulus 1 $(A_j - A_i)$	Stimulus 2 $(G_k - A_i)$
Dehuai et al. [22]	$\left[\cos(\theta_j^o - \theta_i^o), \cos(\theta_j^g - \theta_i^g)\right]$	$\left[\cos(\theta_i^o - \theta_j^o), \cos(\theta_i^g - \theta_j^g)\right]$	potential functions

Table ended

Appendix B

List of Symbols

Notation	Description
A_{f_i}	Affinity of ith selected antibody
d_i	Distance between ith antibody and antigen
c	Scaling factor for auxiliary data
β_i	Auxiliary data for ith selected antibody
μ_i	Antibody maturation rate
K_1, K_2	Maturation constant $<$ best affinity factor and Maturation decay factor
C_i	Clone of ith selected antibody
γ	Scaling factor for random number generator
A_i	The ith Antibody
G_k	The kth Antigen
N_a, N_g	Number of antibodies and antigens, respectively
a_i	Concentration of ith antibody
y_k	Concentration of kth antigen
m_{ij}	Matching function between ith and jth antibodies
n_{ik}	Affinity between ith antibody and jth antigen
α_a, α_s	Stimulation rate, Suppression rate
λ_i	Antibody death rate
O_p	The pth output of DCA
W_X	Weights of different DCA signals
P_i	Input signal categorized as PAMP
D_i	Input signal categorized as Danger
S_i	Input signal categorized as Safe
β	Inflammation level ($\beta \geq 1$)
$\mathbf{B}$	Bacteria matrix with n bacteria ($\mathbf{b}$)
$\mathbf{M}$	Monocyte matrix with n monocytes ($\mathbf{m}$)
$\mathbf{C}$	Cytokine matrix with n cytokines ($\mathbf{c}$)
$\mathbf{R}$	Rotation matrix
$\mathbf{p}$	Current position in cartesian coordinates
C_{xxx}	Concentration of chemoattractants/chemorepellents
$N_{(i,j)}$	Neighborhood of location (i, j)
Pr	Probability of moving from current to next position
I	Inflammation level
κ	Collision factor
$\mathbf{P}$	Pathogen associated molecular pattern signals
$\mathbf{D}$	Danger signals
$\mathbf{S}$	Safe signals
$d(\theta)$	Sensed distance information in θ
θ_F, θ_T	Frontal direction of robot
$\mathbf{O}$	Output for dendritic cell maturity
$\mathbf{T}$	T lymphocyte matrix
E_f, E_r, E_m, E_ω	Rewards/penalities for neighborhood-energy
φ, γ, μ, ω	Energy factors for food, good move, normal move and wait, respectively
C_c	Penalty for collision
I_i, Pa_i, E_i	Binary component of idiotopes, paratopes and epitopes

Appendix C

List of Abbreviations

Notation	Description
AIS	Artificial Immune System
BIS	Biological Immune System
IBF	Immunity-based Framework
HMRS	Heterogeneous Mobile Robotic Systems
SAR	Search and Rescue
CS	Clonal Selection
IN	Immune Network
DT	Danger Theory
GA	Genetic Algorithm
FS	Fuzzy Systems
RL	Reinforcement Learning
APC	Antigen Presenting Cell
PAMP	Pathogen Associated Molecular Patterns
DCA	Dendritic Cell Algorithm
TLR	Toll Like Receptor
CSM	Co-Stimulatory Molecule
DS	Danger Signal
SS	Safe Signal
LIN	Local Immune Network
ID	Identification
ORM	Obstacle Restriction Method
RIN	Reactive Immune Network
VFH	Vector Field Histogram
NK	Natural Killer cells
LTL	Long Term Learning
STL	Short Term Learning
EC	Event Count
CC	Collision Count
MS	Mission Success

Index

Abstract, vi
Acknowledgments, v
Adaptive Component of the Framework, 76
Adaptive Immunity, 89
AIS representation of HMRS, 54
Algorithm, 69
APC, 66
Appendices, 140

B Lymphocytes, 85
Background and Related Work, 13
Biased Random Walk, 59
Bibliography, 168

chemoattractant, 121
chemoattractants, 66
chemorepellent, 121
chemorepellents, 66
Chemotaxis, 58
Clonal Selection, 85
Computational Environment, 129
Conclusions and Future Directions, 135

danger, 63
Danger signals, 64
Danger theory, 63
Dedication, iv
Dendritic cells, 63
Details of Immuno-inspired Robotic Applications, 141
Distributed-world Scenarios, 99, 118, 127

Experiment Scenarios, 94
Experimentation, 93

heterogeneity, 104
Heterogeneous Mobile Robotic Systems, 121
HMRS, 121

Idiotypic Network, 86
Immune Memory, 106
Immune-memory, 88
Inflammation, 61
Initiation of T-cell Maturity, 66
Innate Component of the Framework, 51
Innate Immunity Module, 66
Introduction, 1

Kernel density estimation, 60

Abbreviations, 150
List of Symbols, 149

Mapped-world Scenarios, 96, 110
Maze Navigation, 113
Maze-world Scenarios, 98, 113
Metrics, 104
mission-success, 106
Monte Carlo, 59, 69

Phagocytosis, 57

Reactive Immune Network, 132
Representation, 79
Results and Discussion, 110

Robot configurations, 102

SAR, 110
Search and Rescue, 110
Shaped-world Scenarios, 95, 121
Simulator, 106
Single-Robotic Systems, 110

T Lymphocytes, 83

Vector Field Histogram, 131

Bibliography

[1] *Understanding the Immune System: How It Works*. National Institute of Allergy and Infectious Diseases, September 2007.

[2] U. Aickelin, P. Bentley, S. Cayzer, J. Kim, and J. McLeod. Danger theory: The link between ais and ids? *Artificial Immune Systems*, pages 147–155, 2003.

[3] Wolgang Alt. Biased random walk models for chemotaxis and related diffusion approximations. *Journal of Mathematical Biology*, 9:147–177, 1980.

[4] Tucker Balch. *Behavioral Diversity in Learning Robot Teams*. PhD thesis, Georgia Institute of Technology, 1998.

[5] Gianluca Baldassarre, Stefano Nolfi, and Domenico Parisi. Evolving mobile robots able to display collective behaviors. *Artificial Life*, 9(3):255–267, 2003.

[6] Margherita Barile and Eric W. Weisstein. Neighborhood.

[7] J. Barraquand, B. Langlois, and J.-C. Latombe. Numerical potential field techniques for robot path planning. *Systems, Man and Cybernetics, IEEE Transactions on*, 22(2):224 –241, mar/apr 1992.

[8] R. E. Billngham, L. Brent, and P. B. Medawar. /'actively acquired tolerance/' of foreign cells. *Nature*, 172(4379):603–606, October 1953.

[9] J. Borenstein and Y. Koren. Real-time obstacle avoidance for fact mobile robots. *Systems, Man and Cybernetics, IEEE Transactions on*, 19(5):1179 –1187, sep/oct 1989.

[10] J. Borenstein and Y. Koren. The vector field histogram-fast obstacle avoidance for mobile robots. *Robotics and Automation, IEEE Transactions on*, 7(3):278 –288, jun 1991.

[11] Peter Bretscher and Melvin Cohn. A theory of self-nonself discrimination. *Science*, 169(3950):1042–1049, 1970.

[12] R. Brooks. A robust layered control system for a mobile robot. *Robotics and Automation, IEEE Journal of*, 2(1):14 – 23, mar 1986.

[13] F. M. Burnet. *The Clonal Selection: Theory of Acquired Immunity*. Vanderbilt University Press Tennessee, 1959.

[14] Daniele Calisi and Daniele Nardi. Performance evaluation of pure-motion tasks for mobile robots with respect to world models. *Autonomous Robots*, 27(4):465–481, 2009.

[15] Leandro N. d. Castro. *Artificial Immune Systems: A New Computational Intelligence Approach*. Springer-Verlag, London, 2002.

[16] Tejbanta Chingtham and Shivashankar Nair. Modeling a multiagent mobile robotics test bed using a biologically inspired artificial immune system. *Multi-Agent Systems for Society*, pages 270–283, 2009.

[17] Vincenzo Cutello, Giuseppe Nicosia, Mario Pavone, and Giovanni Stracquadanio. An information-theoretic approach for clonal selection algorithms. In Emma Hart, Chris McEwan, Jon Timmis, and Andy Hone, editors, *Artificial Immune Systems*, volume 6209 of *Lecture Notes in Computer Science*, pages 144–157. Springer Berlin / Heidelberg, 2010.

[18] Vincenzo Cutello and Mario Romeo. On the convergence of immune algorithms. In *In Proceedings of the 1st IEEE Symposium on Foundations of Computational Intelligence*, 2007.

[19] Dipankar Dasgupta and Luis Fernando Nino. *Immunological Computation: Theory and Applications*. Auerbach Publications, 2009.

[20] Rob De Boer and Pauline Hogeweg. Stability of symmetric idiotypic networksa critique of hoffmann's analysis. *Bulletin of Mathematical Biology*, 51(2):217–222, March 1989.

[21] Rob J. de Boer and Alan S. Perelson. Size and connectivity as emergent properties of a developing immune network. *Journal of Theoretical Biology*, 149(3):381 – 424, 1991.

[22] Zeng Dehuai, Xu Gang, Xie Cunxi, and Yu degui. Artificial immune algorithm based robot obstacle-avoiding path planning. In *Automation*

and Logistics, 2008. ICAL 2008. IEEE International Conference on, pages 798 –803, 1-3 2008.

[23] Q.J. Duan, R.X. Wang, H.S. Feng, and L.G. Wang. Applying synthesized immune networks hypothesis to mobile robots. In *Autonomous Decentralized Systems, 2005. ISADS 2005. Proceedings*, pages 69 – 73, 4-8 2005.

[24] J. D. Farmer, N. H. Packard, and A. S. Perelson. The immune system, adaptation, and machine learning. *Phys. D*, 2(1-3):187–204, 1986.

[25] J.L. Fernandez, R. Sanz, J.A. Benayas, and A.R. Dieguez. Improving collision avoidance for mobile robots in partially known environments: the beam curvature method. *Robotics and Autonomous Systems*, 46(4):205 – 219, 2004.

[26] Simon Garrett. A paratope is not an epitope: Implications for immune network models and clonal selection. *Artificial Immune Systems*, pages 217–228, 2003.

[27] Julie Greensmith and Uwe Aickelin. The deterministic dendritic cell algorithm. *Artificial Immune Systems*, pages 291–302, 2008.

[28] Julie Greensmith and Uwe Aickelin. Artificial dendritic cells: Multi-faceted perspectives. In Andrzej Bargiela and Witold Pedrycz, editors, *Human-Centric Information Processing Through Granular Mod-*

elling, volume 182 of *Studies in Computational Intelligence*, pages 375–395. Springer Berlin / Heidelberg, 2009.

[29] Julie Greensmith, Uwe Aickelin, and Jamie Twycross. Articulation and clarification of the dendritic cell algorithm. In Hugues Bersini and Jorge Carneiro, editors, *Artificial Immune Systems*, volume 4163 of *Lecture Notes in Computer Science*, pages 404–417. Springer Berlin / Heidelberg, 2006.

[30] Xuanzi Hu. Clonal selection based mobile robot path planning. In *Automation and Logistics, 2008. ICAL 2008. IEEE International Conference on*, pages 437 –442, 1-3 2008.

[31] Xuanzi Hu and Qingui Xu. Robot path planning based on artificial immune network. In *Robotics and Biomimetics, 2007. ROBIO 2007. IEEE International Conference on*, pages 1053 –1057, 2007.

[32] Nirmal Hui and Dilip Pratihar. Soft computing-based navigation schemes for a real wheeled robot moving among static obstacles. *Journal of Intelligent and Robotic Systems*, 51(3):333–368, March 2008.

[33] Jaeho Hur. *Multi-robot system control using artificial immune system*. PhD thesis, University of Texas at Austin, 2007.

[34] A. Ishiguro, S. Kuboshiki, S. Ichikawa, and Y. Uchikawa. Gait coordination of hexapod walking robots using mutual-coupled immune networks.

In *Proc. IEEE International Conference on Evolutionary Computation*, volume 2, pages 672–677, 29 Nov.–1 Dec. 1995.

[35] A. Ishiguro, Y. Watanabe, T. Kondo, Y. Shirai, and Y. Uchikawa. A robot with a decentralized consensus-making mechanism based on the immune system. In *Autonomous Decentralized Systems, 1997. Proceedings. ISADS 97., Third International Symposium on*, pages 231 –237, April 1997.

[36] Akio Ishiguro, Toshiyuki Kondo, Yuji Watanabe, and Yoshiki Uchikawa. Immunoid: An immunological approach to decentralized behavoir arbitration of autonomous mobile robots. In *PPSN IV: Proceedings of the 4th International Conference on Parallel Problem Solving from Nature*, pages 666–675, London, UK, 1996. Springer-Verlag.

[37] Charles A. Janeway. How the immune system works to protect the host from infection: A personal view. *Proceedings of the National Academy of Sciences*, 98(13):7461–7468, 2001.

[38] N. K. Jerne. Towards a network theory of the immune system. *Annales d'immunologie*, 125C(1-2):373–389, January 1974.

[39] Jin-Hyung Jun, Dong-Wook Lee, and Kwee-Bo Sim. Realization of cooperative strategies and swarm behavior in distributed autonomous robotic systems using artificial immune system. In *Systems, Man, and Cybernetics, 1999. IEEE SMC '99 Conference Proceedings. 1999 IEEE International Conference on*, volume 6, pages 614 –619 vol.6, 1999.

[40] Andreas Kolling, Behzad Tabibian, Arnoud Visser, and Stephen Balakirsky. Robocup rescue simulation league: Virtual robots competition. *RoboCup 2011*, pages 1–5, 2011.

[41] Lukas König, Sanaz Mostaghim, and Hartmut Schmeck. Online and onboard evolution of robotic behavior using finite state machines. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems - Volume 2*, AAMAS '09, pages 1325–1326, Richland, SC, 2009. International Foundation for Autonomous Agents and Multiagent Systems.

[42] Y. Koren and J. Borenstein. Potential field methods and their inherent limitations for mobile robot navigation. In *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*, pages 1398 –1404 vol.2, apr 1991.

[43] Michael Krautmacher and Werner Dilger. Ais based robot navigation in a rescue scenario. *Artificial Immune Systems*, pages 106–118, 2004.

[44] K J Lafferty and A J Cunningham. A new analysis of allogeneic interactions. *Aust J Exp Biol Med*, 53(1):27–42, February 1975.

[45] R. E. Langman and M. Cohn. The 'complete' idiotype network is an absurd immune system. *Immunology Today*, 7(4):100 – 101, 1986.

[46] Dong Wook Lee and Kwee Bo Sim. Artificial immune network-based cooperative control in collective autonomous mobile robots. In *Robot and*

Human Communication, 1997. RO-MAN '97. Proceedings., 6th IEEE International Workshop on, pages 58 –63, sep-1 oct 1997.

[47] Jian Hua Li and Sun An Wang. Model of immune agent and application in path finding of autonomous robots. In *Proc. International Conference on Machine Learning and Cybernetics*, volume 3, pages 1961–1964, 2–5 Nov. 2003.

[48] Meiyi Li, Zixing Cai, Yuexiang Shi, and Pingan Gao. A hybrid immune evolutionary computation based on immunity and clonal selection for concurrent mapping and localization. *Advances in Natural Computation*, pages 1308–1311, 2005.

[49] Guan-Chun Luh and Wei-Chong Cheng. Behavior-based intelligent mobile robot using an immunized reinforcement adaptive learning mechanism. *Advanced Engineering Informatics*, 16(2):85 – 98, 2002.

[50] Guan Chun Luh and Wei Wen Liu. An immunological approach to mobile robot reactive navigation. *Appl. Soft Comput.*, 8(1):30–45, 2008.

[51] Guan-Chun Luh, Chun-Yin Wu, and Wei-Wen Liu. Artificial immune system based cooperative strategies for robot soccer competition. In *Strategic Technology, The 1st International Forum on*, pages 76 –79, 18-20 2006.

[52] M. H. Mabrouk and C. R. McInnes. Solving the potential field local minimum problem using internal agent states. *Robot. Auton. Syst.*,

56(12):1050–1060, 2008.

[53] P. Matzinger. The danger model: a renewed sense of self. *Science*, 296(5566):301–305, April 2002.

[54] Polly Matzinger. Friendly and dangerous signals: is the tissue in control? *Nat Immunol*, 8(1):11–13, January 2007.

[55] Rodney Meyer. Maze. http://www.mathworks.com/matlabcentral, 2005.

[56] R. Michelan and F. J. Von Zuben. Decentralized control system for autonomous navigation based on an evolved artificial immune network. In *CEC '02: Proceedings of the Evolutionary Computation on 2002. CEC '02. Proceedings of the 2002 Congress*, pages 1021–1026, Washington, DC, USA, 2002. IEEE Computer Society.

[57] J. Minguez and L. Montano. Nearness diagram navigation (nd): a new real time collision avoidance approach. In *Intelligent Robots and Systems, 2000. (IROS 2000). Proceedings. 2000 IEEE/RSJ International Conference on*, volume 3, pages 2094 –2100 vol.3, 2000.

[58] N. Mitsumoto, T. Fukuda, F. Arai, and H. Ishihara. Control of the distributed autonomous robotic system based on the biologically inspired immunological architecture. In *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, volume 4, pages 3551 –3556 vol.4, apr 1997.

[59] N. Mitsumoto, T. Fukuda, F. Arai, H. Tadashi, and T. Idogaki. Self-organizing multiple robotic system (a population control through biologically inspired immune network architecture). *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, 2:1614–1619 vol.2, Apr 1996.

[60] Kenneth Murphy, Paul Travers, and Mark Walport. *Janeway's Immunobiology*, volume 7. Garland Science, 2008.

[61] Walter Nowak, Alexey Zakharov, Sebastian Blumenthal, and Erwin Prassler. Benchmarks for mobile manipulation and robust obstacle avoidance and navigation. Technical report, Best Practice in Robotics (BRICS), April 2010.

[62] Robert Oates, Julie Greensmith, Uwe Aickelin, Jonathan Garibaldi, and Graham Kendall. The application of a dendritic cell algorithm to a robotic classifier. *Artificial Immune Systems*, pages 204–215, 2007.

[63] W.J. Opp and F. Sahin. An artificial immune system approach to mobile sensor networks and mine detection. In *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, volume 1, pages 947 –952 vol.1, 0-0 2004.

[64] L. Pallottino, V.G. Scordio, A. Bicchi, and E. Frazzoli. Decentralized cooperative policy for conflict resolution in multivehicle systems. *Robotics, IEEE Transactions on*, 23(6):1170 –1183, dec. 2007.

[65] Lynne E. Parker. *Heterogeneous Multi-Robot Cooperation.* PhD thesis, MIT, 1994.

[66] Lynne E. Parker. Multiple mobile robot systems. In Bruno Siciliano and Oussama Khatib, editors, *Springer Handbook of Robotics*, pages 921–941. Springer Berlin Heidelberg, 2008.

[67] Mathew D. Powers. *Applying inter-layer conflict resolution to hybrid robot control architectures.* PhD thesis, Georgia Institutue of Technology, 2010.

[68] C.E. Prieto, F. Nino, and G. Quintana. A goalkeeper strategy in robot soccer based on danger theory. In *Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence). IEEE Congress on*, pages 3443 –3447, 1-6 2008.

[69] R. Y. Rubinstein and D. P. Kroese. *Simulation and the Monte Carlo Method.* John Wiley & Sons., 2nd ed. edition, 2007.

[70] Erol Sahin. Swarm robotics: From sources of inspiration to domains of application. In Erol Sahin and William Spears, editors, *Swarm Robotics*, volume 3342 of *Lecture Notes in Computer Science*, pages 10–20. Springer Berlin / Heidelberg, 2005.

[71] S. Sathyanath and F. Sahin. Application of artificial immune system based intelligent multi agent model to a mine detection problem. In

Systems, Man and Cybernetics, 2002 IEEE International Conference on, volume 3, page 6 pp. vol.3, oct. 2002.

[72] Lynda M. Stuart and R. Alan Ezekowitz. Phagocytosis and comparative innate immunity: learning on the fly. *Nat Rev Immunol*, 8(2):131–141, February 2008.

[73] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics*. MIT Press, 2005.

[74] J. Timmis, A. Hone, T. Stibor, and E. Clark. Theoretical advances in artificial immune systems. *Theoretical Computer Science*, 403(1):11 – 32, 2008.

[75] Diana Tsankova, Velichka Georgieva, Frantisek Zezulka, and Zdenek Bradac. Immune network control for stigmergy based foraging behaviour of autonomous mobile robots. *Int. J. Adapt. Control Signal Process.*, 21(2-3):265–286, 2007.

[76] Elio Tuci, Christos Ampatzis, Federico Vicentini, and Marco Dorigo. Evolving homogeneous neurocontrollers for a group of heterogeneous robots: Coordinated motion, cooperation, and acoustic communication. *Artificial Life*, 14(2):157–178, 2008.

[77] Jamie Twycross. *Integrated Innate and Adaptive Artificial Immune Systems applied to Process Anomaly Detection*. PhD thesis, University of Nottingham, January 2007.

[78] I. Ulrich and J. Borenstein. Vfh*: local obstacle avoidance with look-ahead verification. In *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, volume 3, pages 2505 –2511 vol.3, 2000.

[79] P.A. Vargas, L.N. de Castro, R. Michelan, and F.J. Von Zuben. Implementation of an immuno-genetic network on a real khepera ii robot. In *Evolutionary Computation, 2003. CEC '03. The 2003 Congress on*, volume 1, pages 420 – 426 Vol.1, 2003.

[80] Patrcia Amncio Vargas, Leandro Nunes de Castro, Roberto Michelan, and Fernando J. Von Zuben. An immune learning classifier network for autonomous navigation. In Jon Timmis, Peter J. Bentley, and Emma Hart, editors, *ICARIS*, volume 2787 of *Lecture Notes in Computer Science*, pages 69–80. Springer, 2003.

[81] M.P. Wand and M.C. Jones. *Kernel Smoothing*. Chapman & Hall/CRC, 1995.

[82] Jijun Wang and Michael Lewis. Assessing cooperation in human control of heterogeneous robots. In *Proceedings of the 3rd ACM/IEEE international conference on Human robot interaction*, HRI '08, pages 9–16, New York, NY, USA, 2008. ACM.

[83] Lei Wang and B. Hirsbrunner. An evolutionary algorithm with population immunity and its application on autonomous robot control. In

Evolutionary Computation, 2003. CEC '03. The 2003 Congress on, volume 1, pages 397 – 404 Vol.1, 8-12 2003.

[84] Yen-Nien Wang, Tsai-Sheng Lee, and Teng-Fa Tsao. Plan on obstacle-avoiding path for mobile robots based on artificial immune algorithm. In Derong Liu, Shumin Fei, Zeng-Guang Hou, Huaguang Zhang, and Changyin Sun, editors, *Advances in Neural Networks ISNN 2007*, volume 4491 of *Lecture Notes in Computer Science*, pages 694–703. Springer Berlin / Heidelberg, 2007.

[85] Y. Watanabe, A. Ishiguro, Y. Shirai, and Y. Uchikawa. Emergent construction of behavior arbitration mechanism based on the immune system. In *Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on*, pages 481 –486, may 1998.

[86] A.M. Whitbrook, U. Aickelin, and J.M. Garibaldi. Idiotypic immune networks in mobile-robot control. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 37(6):1581 –1598, dec. 2007.

[87] Amanda Whitbrook, Uwe Aickelin, and Jonathan Garibaldi. Genetic-algorithm seeding of idiotypic networks for mobile-robot navigation. *Proceedings of the International Conference on Informatics in Control, Automation and Robotics (ICINCO 2008), in print, pp, Funchal, Portugal, 2008*, (arXiv:0803.1626), Mar 2008.

[88] Amanda Whitbrook, Uwe Aickelin, and Jonathan Garibaldi. An idiotypic immune network as a short-term learning architecture for mobile robots. In Peter Bentley, Doheon Lee, and Sungwon Jung, editors, *Artificial Immune Systems*, volume 5132 of *Lecture Notes in Computer Science*, pages 266–278. Springer Berlin / Heidelberg, 2008.

[89] Amanda M. Whitbrook, Uwe Aickelin, and Jonathan M. Garibaldi. Two-timescale learning using idiotypic behaviour mediation for a navigating mobile robot. *Applied Soft Computing*, 10(3):876 – 887, 2010.

[90] Jennifer White and Simon Garrett. Improved pattern recognition with artificial clonal selection? *Artificial Immune Systems*, pages 181–193, 2003.

[91] W.L. Xu. A virtual target approach for resolving the limit cycle problem in navigation of a fuzzy behaviour-based mobile robot. *Robotics and Autonomous Systems*, 30(4):315 – 324, 2000.

[92] Li Xuemei, Kam-Cheung Kwong, Li Jincheng, Jiang Liangzhong, and Zeng Dehuai. Mobile robot path planning based on artificial immune algorithm. In *Advanced Robotics and its Social Impacts (ARSO), 2009 IEEE Workshop on*, pages 1 –5, 2009.

[93] Jonathan W. Yewdell and Brian P. Dolan. Immunology: Cross-dressers turn on t cells. *Nature*, 471(7340):581–582, March 2011.

[94] Yanduo Zhang and Tao Lu. Research on cooperative strategies of soccer robots based on artificial immune system. In *Computational Intelligence and Industrial Application, 2008. PACIIA '08. Pacific-Asia Workshop on*, volume 2, pages 656 –659, 2008.

[95] Xi-yong Zou and Jing Zhu. Virtual local target method for avoiding local minimum in potential field based robot navigation. *Journal of Zhejiang University - Science A*, 4(3):264–269, May 2003.